Universität Stuttgart

Fakultät Informatik, Elektrotechnik
und Informationstechnik

Diplomarbeit Nr. 3013

Effiziente automatisierte Erstellung von Straßenkarten

Patrick Baier

Studiengang:	Wirtschaftsinformatik Master
Prüfer:	Prof. Dr. Kurt Rothermel
Betreuer:	Harald Weinschrott
begonnen am:	15. Februar 2010
beendet am:	16. August 2010
CR-Nummer:	C.2.1, C.2.4, C.3, E.1

IPVS

Institut für Parallele
und Verteilte Systeme
Abteilung Verteilte Systeme
Universitätsstraße 38
D-70569 Stuttgart

NEXUS

Zusammenfassung

Die vorliegende Masterarbeit stellt einen Ansatz zur effizienten, automatisierten Erstellung von Straßenkarten mit Hilfe von GPS-Sensordaten vor. Die dafür benötigten GPS-Daten werden von gewöhnlichen Personen gesammelt, welche ihre Mobilgeräte wie gewohnt nutzen, aber auf ihren täglichen Wegen zusätzlich GPS-Positionsinformationen erfassen, welche sie dem System zur Verfügung stellen. Durch Verarbeitung dieser Daten, erstellt dieses eine Straßenkarte, die alle erfassten Daten zu einem eindeutigen Straßennetz integriert.

Da die Personen, welche ihre GPS-Daten dem System beisteuern, dafür nicht direkt entlohnt werden, muss das System bei diesen Personen eine hohe Akzeptanz genießen, damit genügend Datenquellen für die Erstellung einer qualitativ hochwertigen Straßenkarte zur Verfügung stehen. Dazu ist es wichtig, dass der Ressourcenverbrauch der Mobilgeräte, der durch die Erfassung und Übertragung der GPS-Daten entsteht, möglichst minimal ist, damit für die teilnehmenden Personen durch das Erfassen der GPS-Daten kein überhöhter Ressourcenverbrauch spürbar ist. Dies erfordert einen Algorithmus, der die Erfassung der GPS-Daten durch die Mobilgeräte koordiniert.

Um diesen Herausforderungen zu begegnen, muss im ersten Schritt ein Verfahren gefunden werden, welches die gesammelten GPS-Daten zu einer einheitlichen Straßenkarte integriert und dabei die inhärente Unschärfe von GPS-Ortungen berücksichtigt. Im zweiten Schritt muss ein Algorithmus entwickelt werden, welcher die Mobilgeräte so steuert, dass möglichst wenig redundante Erfassungen entstehen, damit die Ressourcen dieser Geräte geschont werden. Um dies zu erreichen, sollen die Mobilgeräte nur die Bereiche erfassen, welche in der bisher erstellten Straßenkarte qualitativ noch nicht ausreichend verfügbar sind. Damit diese Entscheidung getroffen werden kann, muss eine Qualitätsmetrik entwickelt werden, welche eine Aussage über die Qualität der erfassten Daten ermöglicht.

Diese Masterarbeit stellt ein solches Verfahren vor und bietet Lösungen für die oben beschriebenen Herausforderungen. Dieses wird zuerst konzeptionell entworfen und anschließend in geeigneter Weise implementiert. Abschließend wird im Rahmen einer Evaluation gezeigt, dass durch Anwendung der entworfenen Konzepte die relative Einsparung an Ressourcen höher ist als die dadurch entstandene Qualitätsminderung bei der erstellten Straßenkarte.

Inhaltsverzeichnis

Abbildungsverzeichnis **iv**

Tabellenverzeichnis **vi**

1 Einleitung **1**

1.1 Motivation 1

1.2 Zielsetzung 3

1.3 Aufbau 3

2 Grundlagen **5**

2.1 Übersicht 5

2.2 Urban Sensing 6

2.2.1 Grundlagen 6

2.2.2 Möglichkeiten der Teilnehmerpartizipation 8

2.2.3 Technische Aspekte 8

2.2.4 Anwendungsbeispiele 11

2.3 Collaborative Mapping 12

2.3.1 Grundlagen 12

2.3.2 Auswirkungen für die Kartographie 13

2.3.3 Das Global Positioning System 14

2.3.4 Anwendungsbeispiele 16

2.4 Stand der Forschung 17

2.4.1 Verfahren zur Analyse von GPS-Tracks 18

2.4.2 Beitrag dieser Arbeit 19

3 Konzeptionelle Umsetzung **21**
3.1 Systemmodell . . . 21
3.1.1 Weltmodell . . . 22
3.1.2 GPS-Fehlermodell . . . 23
3.1.3 Hardwarekomponenten . . . 24
3.2 Straßenerfassung . . . 25
3.2.1 Datenmodelle und -strukturen . . . 25
3.2.2 Erfassung von GPS-Tracks . . . 27
3.3 Qualitätsmodell . . . 36
3.3.1 Qualität einzelner Straßen . . . 37
3.3.2 Zeitwert der Erfassung . . . 43
3.3.3 Qualität einzelner Gitterzellen . . . 47
3.4 Steuerung der Erfassung . . . 48
3.4.1 Verwaltung der Mobilgeräte . . . 48
3.4.2 Lokationsmanagement . . . 49
3.4.3 Unterteilung des Zielbereichs . . . 50
3.4.4 Initialisierung der Erfassung . . . 51
3.4.5 Übertragung der GPS-Daten . . . 52
3.4.6 Erkennung von neuen Geräten . . . 54
3.4.7 Statusänderungen bei Gitterzellen . . . 57
3.4.8 Beispielszenario . . . 58
3.4.9 Kategorisierung von Straßen . . . 59
3.4.10 Optimierungspotentiale . . . 62
3.5 Zusammenfassung . . . 65

4 Implementierung **66**
4.1 Der ns-2 Netzwerksimulator . . . 66
4.1.1 Grundlagen . . . 67
4.1.2 OTcl . . . 67
4.1.3 C++ . . . 68
4.2 Umsetzung der Konzepte . . . 69
4.2.1 Allgemeines . . . 69

4.2.2 Grundlegende Architektur 70
4.2.3 Ablaufdiagramme 72
4.2.4 Grenzen der Implementierung 74
4.3 Zusammenfassung 75

5 Evaluation **76**
5.1 Tracefiles 76
5.1.1 Grundlagen 77
5.1.2 Erstellung von Tracefiles 78
5.2 Parameter 79
5.2.1 Eingabeparameter 79
5.2.2 Ausgabewerte des Systems 80
5.3 Simulation 82
5.3.1 Standardszenario 83
5.3.2 Simulation 83
5.4 Zusammenfassung der Ergebnisse 91

6 Zusammenfassung **93**
6.1 Ausblick 94

Abbildungsverzeichnis

3.1 Abbildung vom WGS84-Bezugsystem auf 2D-Fläche 22
3.2 Konfidenzintervall für σ 24
3.3 Zielbereich mit Gitterzellen der Seitenlänge g 26
3.4 Mögliche Probleme beim Vergleich von zwei Tracks 29
3.5 Erstellung von Histogrammen 30
3.6 Berechnung des Wertes eines Plateaus 31
3.7 Teilung eines GPS-Tracks beim Vergleich mit einer Straße 32
3.8 Interpolation der Punkte zweier GPS-Tracks 33
3.9 Verknüpfung zwischen GPS-Track und Gitterzellen 34
3.10 Mögliche Positionen bei der Bestimmung der Straßenbreite 35
3.11 Problemfall bei der Bestimmung der Straßenbreite 36
3.12 Kantenwinkel bei der Genauigkeit des Kurvenverlaufs 38
3.13 Definition des Winkel α 39
3.14 Resultierende Straßenqualität bei Verwendung der Produkt-Funktion . . . 42
3.15 Zeitwert-Funktion für verschiedene Steigungsparameter 46
3.16 Ablauf der Initialisierung 53
3.17 Verbindung eines Tracks mit einer Straße 54
3.18 Definition eines Randbereichs für einzelne Zellen 55
3.19 Vereinfachung eines GPS-Tracks 64

4.1 Interaktion des ns-2 Simulators 68
4.2 Klassendiagramm der Systemarchitektur 70
4.3 Sequenzdiagramm zur Verarbeitung eines neuen Tracks 73
4.4 Positionserfassung der Mobilgeräte 74

5.1 Anteil der Wegpunkte und #GPS-Ortungen 85
5.2 Anzahl der durchgeführten Fusionen und Straßenqualität im Vergleich . . . 86
5.3 Energieverbrauch der Mobilgeräte 88
5.4 Vergleich Abnahme Qualität zu Einsparung Ressourcen 90

Tabellenverzeichnis

3.1 Straßenklassen 60

5.1 Parameterbelegung für Simulation 78
5.2 Eingabeparameter für Simulation 79
5.3 Energieverbrauch eines Mobilgeräts 81
5.4 Parameterbelegung Standardszenario 83

Kapitel 1

Einleitung

1.1 Motivation

Vor ungefähr zwanzig Jahren veränderte sich die Art und Weise, wie Menschen miteinander kommunizieren, drastisch. Mit der Einführung des Mobiltelefons und der dafür benötigten Infrastruktur, war es plötzlich möglich Telefonate von nahezu jedem beliebigen Ort aus zu führen und dauerhaft erreichbar zu sein. Seit dieser Zeit unterlag die Mobilfunktechnik einer ständigen technischen Weiterentwicklung, welche die klassische Anwendung der drahtlosen Telefonie mittlerweile bei weitem übersteigt. Der aktuelle Trend geht in den letzten Jahren hin zu so genannten Smartphones. Das sind Mobiltelefone, die durch ein eigenes Betriebssystem und leistungsfähiger Hardware eine Vielzahl von Anwendungen ausführen können und somit den Eigenschaften eines Mini-Computers gleichzusetzen sind. Weiterentwicklungen in der Übertragungstechnik der Mobilfunktechnik machen es zudem möglich, Datenpakete über das Mobilfunknetz mit einer akzeptablen Datenrate zu versenden [24]. Kombiniert mit entsprechenden Flatrate-Tarifen der Netzbetreiber, ermöglichten diese Fortschritte den Durchbruch des mobilen Internets für den Endanwender.

Neben diesen Entwicklungen in der Mobilfunktechnik, gibt es in letzter Zeit einen weiteren Trend, der die Einbettung von Sensoren in Smartphones forciert, so dass mittlerweile die Mehrheit der erhältlichen Mobiltelefone Sensoren enthält, mit denen Umwelt- und Positionsinformationen gesammelt und ausgewertet werden können. Beispiele für solche Sensoren sind ein GPS-Empfänger oder ein Kompass. Die Tatsache, dass tagtäglich eine Vielzahl dieser Sensoren auf Mobilgeräten ungenutzt von Menschen mitgeführt werden, eröffnet eine neue Dimension für Anwendungen, welche diese Sensordaten verarbeiten könnten. So wäre es ohne zusätzliche Kosten möglich, dieses Angebot an Datenquellen zu nutzen, um Informationen über Geografie und Umweltzustände zu erfassen, zu verarbeiten und für die verschiedensten Zwecke aufbereitet zur Verfügung zu stellen. Diese neue Herausforderung

findet in letzter Zeit unter dem Namen *Urban Sensing* [14] zunehmende Beachtung, da ihr Potential eine Reihe von dynamischen Beobachtungen ermöglicht, welche bisher nur ansatzweise durch den Einsatz von statischen Sensoren möglich waren.

Einer der am meisten genutzten Sensoren in diesem Kontext ist der GPS-Empfänger, welcher die Bestimmung der aktuellen Position des Mobilgeräts ermöglicht. Dadurch konnten bereits die ersten Navigationssysteme auf Smartphones erfolgreich umgesetzt werden, indem sie die via GPS ermittelte Positionsinformationen auf einer im System vorhandenen Straßenkarte darstellen und mit Hilfe von Routingalgorithmen den Weg zu einem zuvor festgelegten Ziel berechnen. Im Gegensatz dazu, können die erfassten GPS-Daten nicht nur zur Orientierung auf Karten sondern auch zur Erstellung von Karten genutzt werden. Einen ersten erfolgreichen Ansatz bietet hierzu das Projekt *OpenStreetMaps* (OSM)[22]. Bei OSM zeichnen die Benutzer mit ihren GPS-Geräten oder ihren GPS-fähigen Mobiltelefonen gezielt ihre zurückgelegten Strecken in Form von Wegpunkten auf, welche sie anschließend auf den Projektserver laden können. Dadurch entsteht nach und nach eine Straßen- und Landschaftskarte, welche im Internet frei zugänglich und weiterverwendbar ist. OSM bildet somit ein kostenloses Gegenprodukt zum proprietären Kartenmaterial der großen Anbieter des Navigationsmarkts. Da es jedoch auf den aktiven Beitrag seiner Teilnehmer angewiesen ist, gibt es immer noch geografische Gebiete, die bisher gänzlich unerschlossen oder nur in minderwertiger Qualität vorhanden sind. Um diesen Nachtteil zu überwinden, wäre ein System nötig, welches die vorhandenen GPS-Empfänger der Mobilgeräte nutzt, ohne dass eine Interaktion der Teilnehmer nötig ist. Da dies für die Teilnehmer keinen Mehraufwand darstellt, könnten höhere Teilnehmerzahlen und somit eine bessere Qualität der erstellten Karte erreicht werden.

Die Motivation dieser Arbeit besteht darin, ein Konzept zu finden, das den Ansatz des Urban Sensing mit der Idee von OSM verbindet und es so ermöglicht, die Erstellung von Straßenkarten zu automatisieren. Dieses Ziel soll durch die automatische Verarbeitung von GPS-Sensordaten, die von Mobilgeräten erfasst wurden, umgesetzt werden. Im Gegensatz zu OSM soll dies aber ohne manuelles Mitwirken der Teilnehmer geschehen — die Sensordaten sollen automatisch von den Mobilgeräten ausgelesen werden. Ein Teilnehmer trägt sein Mobiltelefon wie gewohnt bei sich, während die von ihm aufgezeichneten GPS-Daten automatisch an eine zentrale Stelle weitergeleitet werden, welche diese Daten analysiert und in ein Gesamtmodell integriert. Der Vorteil gegenüber den bereits existierenden Verfahren liegt darin, dass durch das automatische Auslesen der GPS-Empfänger mehr Datenquellen zur Verfügung stehen und die Kartenerstellung auch in Regionen durchgeführt werden kann, in denen keine Teilnehmer zur manuellen Erstellung bereit sind. Dies steigert zum einen die Geschwindigkeit der Kartenerstellung und zum anderen die Qualität der zu erstellenden Karte.

1.2 Zielsetzung

Ziel der vorliegenden Masterarbeit ist es, ein Verfahren zu entwickeln, welches eine effiziente, automatisierte Erstellung von Straßenkarten ermöglicht. Dieses soll GPS-Daten von mobilen Sensoren sammeln, auswerten und daraus eine Karte erstellen, welche das Straßennetz eines zuvor festgelegten geografischen Bereichs wiedergibt.

Das zu entwerfende System soll als Eingabe einen bisher nicht kartografierten Gebietsausschnitt erhalten und dann, mittels den in diesem Gebiet befindlichen mobilen Geräten und deren GPS-Empfängern, die nötigen Informationen sammeln und verarbeiten, um eine Karte des Straßennetzes zu erstellen. Hierbei müssen die Besitzer der Mobilgeräte nicht aktiv bei der Erstellung der Karte mitwirken, sie sollen lediglich ihr Mobilgerät auf ihren alltäglichen Wegen mit sich führen. Die benötigten GPS-Daten werden dann automatisch an das System übertragen, welches diese zu einer Gesamtkarte integriert.

Da Mobilgeräte nur über eine begrenzte Batterieleistung verfügen und eine hoher Energieverbrauch von den Teilnehmer in den meisten Fällen nicht toleriert wird, sollen die Anzahl der Sende- und GPS-Operationen dieser Geräte auf ein Minimum begrenzt werden, ohne dabei einen allzu drastischen Qualitätsverlust bei der Erstellung der Straßenkarte zu erleiden. Um dies zu erreichen, soll das System die Erfassung der Sensordaten gezielt steuern, so dass Bereiche, die bis zu einem bestimmten Zeitpunkt überdurchschnittlich gut erfasst wurden, von der Erfassung vorübergehend ausgesetzt werden, um dadurch die Ressourcen der darin befindlichen Geräte zu schonen. Damit das System entscheiden kann, welche Bereiche wie gut erfasst wurden, ist die Definition einer Qualitätsmetrik für bereits erfasste Straßen notwendig, welche zuerst entwickelt werden muss.

Um das beschriebene Verfahren realisieren zu können sind zuerst eine Reihe von Teilzielen umzusetzen, welche im folgenden Abschnitt näher beschrieben werden.

1.3 Aufbau

Die vorliegende Masterarbeit gliedert sich in sechs Kapitel, welche von dieser Einführung bis zur abschließenden Zusammenfassung reichen. Beginnend mit diesem Kapitel, welches Motivation und Zielsetzung dieser Arbeit verdeutlich hat, führen die im Folgenden kurz vorgestellten Kapitel zur Umsetzung des vorgestellten Ziels.

Bevor wir damit beginnen, Konzepte zur Lösung der Zielsetzung zu entwerfen, legen wir in Kapitel 2 die Grundlagen, welche den Rahmen, in welcher diese Masterarbeit einzuordnen ist, genauer vorstellen. Dieser umfasst das Teilgebiete Urban Sensing und das der gemeinschaftlichen Erstellung von Karten (*Collaborative Mapping* [19]). Da die Einführung

in diese Themen eher auf eine allgemeine Art erfolgt, wird zum Abschluss des Kapitels der aktuelle Stand der Forschung zur vorliegenden Problemstellung gezielt vorgestellt.

Nachdem die theoretischen Grundlagen und Techniken vorgestellt wurden, werden in Kapitel 3 die selbst erarbeiteten Konzepte präsentiert, welche das Ziel dieser Masterarbeit realisieren sollen. Dazu wird ein Algorithmus entworfen, der die Erfassung der GPS-Sensordaten automatisiert, die Integration dieser Daten zu einer Straßenkarte vornimmt und die GPS-Positionserfassung der Mobilgeräten so koordiniert, dass deren Ressourcenbelastung möglichst minimiert wird, ohne dabei größere Einbußen bei der Qualität der Straßenkarte hinnehmen zu müssen. Um diesen Algorithmus zu realisieren, muss jedoch zuerst ein Verfahren gefunden werden, mit dessen Hilfe sich mehrere ähnliche GPS-Tracks zu einer einheitlichen Straße integrieren lassen. Zudem muss für die erfassten Straßen eine Qualitätsmetrik eingeführt werden, anhand welcher sich beurteilen lässt, wie gut einzelne Bereiche des zu erfassenden Kartenausschnitts bereits erfasst sind.

Nachdem alle nötigen Konzepte definiert wurden, gilt es diese anschließend in geeigneter Form zu implementieren. Kapitel 4 stellt daher die für die Implementierung verwendeten Werkzeuge vor und beschreibt die resultierende Systemarchitektur. Dabei wird der für die nachfolgende Evaluation wichtige Netzwerksimulator ns-2 vorgestellt, auf dessen Basis auch die Implementierung stattfand.

Mit Hilfe des implementierten Systems, können nun die erstellten Konzepte evaluiert werden. Kapitel 5 beschreibt daher, welche Szenarios und Testfälle bei der Evaluation simuliert wurden und welche Ergebnisse dabei erzielt wurden. Es werden unterschiedliche Konfigurationen von Systemparametern untersucht, deren Auswirkungen analysiert und versucht daraus allgemeine Schlüsse zu ziehen.

Abschließend fasst Kapitel 6 die Ergebnisse dieser Masterarbeit noch einmal in übersichtlicher Form zusammen und gibt einen Ausblick darauf, wie diese in künftigen Arbeiten genutzt und weiterentwickelt werden können.

Kapitel 2

Grundlagen

2.1 Übersicht

Die in der Zielsetzung gesetzten Ziele sind inhaltlich mit zwei verwandten Themengebieten verbunden, welche in diesem Grundlagenteil vorgestellt werden sollen, um dem Leser einen Überblick über die Diskussion zu geben, welche diese momentan umgibt. Sie unterteilen sich in die beiden Themen *Urban Sensing* und das darauf aufbauende *Collaborative Mapping*.

Die Erhebung und Verarbeitung von GPS-Daten, welche von einem Mobilgerät stammen, das von einer gewöhnlichen Person in deren Alltag mitgeführt wird, fällt unter das relativ neue Paradigma des Urban Sensing. Dieses nutzt bereits vorhandene aber momentan nicht genutzte Sensoren, wie sie in den meisten heutigen Mobilgeräten zu finden sind, um Informationen zu ermitteln, welche mit statischen Sensoren nur schwer erreichbar wären. Die Nutzung der für die Zielsetzung benötigten mobilen Sensoren zur Erfassung von GPS-Positionen ist eine Anwendung dieser Technik. Daher geht Abschnitt 2.2 auf die Grundlagen und technischen Herausforderungen von Urban Sensing näher ein.

Die gemeinschaftliche Erstellung von Straßenkarten fällt unter den Begriff des Collaborative Mapping. Dieses beschäftigt sich mit der Fragestellung, wie eine Straßenkarte durch mehrere Personen gemeinschaftlich erstellt werden kann. Dabei tauchen Fragen auf, welche von der Bereitstellung bis zur Integration der geografischen Daten reichen. Abschnitt 2.3 stellt die wichtigsten Techniken dieser Methodik vor und bietet erste Lösungsansätze für die Erreichung der Zielsetzung.

Nachdem mit diesen beiden Thematiken die wichtigsten Grundlagen eingeführt wurden, untersucht Abschnitt 2.4 die bisherigen Forschungsergebnisse, welche in Bezug zum konkreten Ziel dieser Masterarbeit stehen. Diese werden vorgestellt und es wird erläutert, wie sie für den folgenden Konzeptteil genutzt werden können.

2.2 Urban Sensing

Die erste Thematik, welche wir in diesem Grundlagenkapitel vorstellen, ist die des Urban Sensing. Dieses Konzept dient zum einen als Grundlage für das in Abschnitt 2.3 folgende Collaborative Mapping und zum anderen gibt es einen allgemeinen Überblick darüber, was momentan im Bereich der Verarbeitung mobiler Sensordaten möglich ist.

Beginnend mit einer Einführung in die Grundlagen und dem Versuch einer allgemeinen Begriffsdefinition, leitet Abschnitt 2.2.1 die Betrachtung von Urban Sensing ein. Nach dieser Einführung werden in Abschnitt 2.2.2 die beiden generellen Arten der Teilnehmerpartizipation des Urban Sensing vorgestellt und die Problemstellung dieser Arbeit entsprechend eingeordnet. Anschließend untersucht Abschnitt 2.2.3 die technischen Komponenten, die für die Umsetzung von Urban Sensing nötig sind und somit in Zusammenhang mit der Lösung der Problemstellung dieser Arbeit stehen. Der abschließende Abschnitt 2.2.4 stellt eine Reihe von aktuellen Projekten vor, die auf dem Ansatz des Urban Sensing basieren und somit einen Eindruck darüber geben, welches Potential dieses Konzept bietet.

2.2.1 Grundlagen

Das Konzept des Urban Sensing ist ein Ansatz, der in aktuellen Forschungsveröffentlichungen erst kürzlich an vermehrter Aufmerksamkeit gewonnen hat, so dass in letzter Zeit eine ganze Reihe von Artikeln, welche die Möglichkeiten und Herausforderungen des Urban Sensing charakterisieren, erschienen sind. Hierbei beschränkt sich die Terminologie allerdings nicht auf einen einheitlichen Begriff, sondern es tauchen neben dem Begriff Urban Sensing (siehe [14],[10],[30]) eine Vielzahl von anderen Begriffen auf, welche im Wesentlichen das gleiche Paradigma beschreiben („Mobiscopes“ [1], „People-Centric Sensing“ [9] „Participatory Sensing“ [8] und „Mobile Sensing“ [26]). Im nachfolgenden Absatz wird daher versucht, aus diesen verschiedenen Quellen, eine einheitliche Definition des Begriffs Urban Sensing abzuleiten.

Definition

Unter dem Begriff Urban Sensing versteht man die Erfassung, Verarbeitung und Auswertung von Sensordaten, welche von Sensoren bereitgestellt werden, die sich auf mobilen Endbenutzergeräten befinden. Diese Sensoren müssen in der Regel nicht beschafft oder platziert werden, sondern sie befinden sich auf Mobilgeräten von gewöhnlichen Personen, welche diese dem Urban Sensing System zur Verfügung stellen. Als Konsequenz hat das System keine Kontrolle über Bewegungsrichtung und Geschwindigkeit dieser Sensoren. Geografisch gesehen eignet sich das urbane Gebiet einer Stadt als ideales Umfeld für die Umsetzung von

Urban Sensing, da die hohe Populationsdichte eine breite Masse an potentiellen Geräten zur Verfügung stellt und somit nahezu das ganze Stadtgebiet sensorisch erfasst werden kann [39]. Weit verbreitete und genutzte mobile Sensoren in diesem Kontext sind Digitalkameras, Mikrofone, Kompasse, GPS-Empfänger, Thermometer und Beschleunigungssensoren, welche auf sehr heterogenen mobilen Geräten platziert sein können.

Vorteile gegenüber statischen Sensornetzwerken

Da beim Urban Sensing mobile Sensoren im Mittelpunkt stehen, welche durch ihre ständigen Bewegungen nicht auf einen starren Ausschnitt der Umwelt fixiert sind, bieten sich für Anwendungen, die Sensordaten verarbeiten, vollkommen neue Potentiale. Diese können durch die dynamische Verteilung der mobilen Endgeräte auf Daten aus Bereichen zurückgreifen, die für statische Sensornetzwerke in diesem Ausmaß nicht zugänglich sind [34]. Dadurch ermöglicht Urban Sensing die Auswertung von Sensordaten für mehrere Anwendungsdomänen zugleich, während statische Sensornetzwerke meist nur für eine spezielle Aufgabe eingesetzt werden können, da damit eine entsprechende Platzierung der Sensoren verbunden ist [39]. Durch die Mobilität der Sensoren besteht außerdem die Möglichkeit, Bereiche durch mehrere Sensoren relativ einfach redundant zu erfassen und somit Ungenauigkeiten einzelner Messungen auszugleichen [2]. Diese Eigenschaften ermöglichen eine enorme Kostenersparnis im Vergleich zur flächendeckend äquivalenten Platzierung von statischen Sensoren [10]. Zusammenfassend lässt sich sagen, dass der Vorteil des Urban Sensing zum einen darin besteht, dass Ressourcen genutzt werden können, die in den meisten Fällen bereits vorhanden aber ungenutzt sind und zum anderen ermöglicht die Mobilität der Sensoren die Bereitstellung von Sensordaten für eine relativ großes geografisches Gebiet, welches mit statischen Sensoren nur schwer erfassbar wäre.

Herausforderungen

Da Urban Sensing noch am Anfang seiner Entwicklung steht, gibt es eine Reihe von Herausforderungen zu bewältigen, bevor eine einheitliche, interoperable und transparente Nutzung der Daten möglich ist, welche es bereitstellt. Diese liegen in den Bereichen der Geräte- und Datenheterogenität, der Privatsphäre der Mobilgerätebesitzer, der Herausforderung einer einheitlichen Netzwerkarchitektur und den sozialen Auswirkungen. Da an dieser Stelle nicht näher auf diese offenen Punkte eingegangen wird, sei auf den Artikel von Abdelzaher et al. [1] verwiesen, welcher eine gute Übersicht über Probleme und mögliche Lösungen bei der Umsetzung von Urban Sensing gibt.

2.2.2 Möglichkeiten der Teilnehmerpartizipation

Je nach Aufgabenbereich und Zielsetzung, kann der Grad an Interaktion der Personen, welche die Mobilgeräte zur Erfassung von Sensordaten mit sich führen (die Teilnehmer), unterschiedlich groß sein. Daher werden beim Urban Sensing, mit dem partizipatorischen bzw. opportunistischen Ansatz, zwei verschiedene Formen der Teilnehmerpartizipation unterschieden, die an dieser Stelle erläutert werden [9].

Beim partizipatorischen Ansatz werden die Teilnehmer bei wichtigen Schritten der Erfassung und Verarbeitung der Sensordaten mit einbezogen. So müssen diese zum Beispiel selbst entscheiden, welche Daten erfasst und an das System weitergeleitet werden sollen. Ihre Motivation liegt in der Regel in einem persönlichen Nutzen, den sie sich von der Anwendung versprechen. Sie befolgen daher aktiv die vom System geforderten Abläufe, um ihre gesammelten Sensordaten beizusteuern und gegebenenfalls weiter anzureichern. Ein Beispiel eines derartigen Ansatzes ist zum Beispiel OpenStreetMaps.

Im Gegensatz dazu sieht der opportunistische Ansatz keine Interaktion der Teilnehmer vor. Diese müssen lediglich die Sensoranwendung auf ihrem Mobilgerät einmalig starten und haben danach in der Regel keinen Einfluss mehr auf die Erfassung und Verarbeitung der Sensordaten, da diese automatisch vom System erhoben werden. Hierbei hat Lane et al. [30] mit Hilfe eines Evaluationsmodells gezeigt, dass der opportunistische Ansatz eine bessere Alternative für großflächige und facettenreichere Anwendungen ist, da durch ihn deutlich mehr Daten bereitgestellt werden. Beim opportunistischen Ansatz ist außerdem zu beachten, dass die Belastung der Mobilgeräte nicht überproportional hoch sein darf, da ansonsten die Akzeptanz durch die Benutzer schwindet und somit die Vorteile eines opportunistischen Ansatzes wieder zunichte gemacht werden.

Wie bereits in der Zielsetzung in Abschnitt 1.2 erwähnt, liegt die Grundidee der vorliegenden Masterarbeit darin, die Erstellung von Straßenkarten zu automatisieren. Diese Forderung verlangt die Umsetzung eines opportunistischen Ansatzes, da auf eine Interaktion der Teilnehmer verzichtet werden soll. Der Grund für diese Forderung liegt in den erwähnten Vorteilen des opportunistischen Ansatzes, welche vorwiegend in der höheren Zahl der erreichbaren Teilnehmer begründet sind. Dadurch kann die Kartenerstellung effizienter und flächendeckender eingesetzt werden. Dies bringt jedoch neue Herausforderungen mit sich, da ein Konzept gefunden werden muss, welches die Ressourcenbelastung der Mobilgeräte möglichst minimiert.

2.2.3 Technische Aspekte

Die Umsetzung von Urban Sensing erfordert eine Reihe von technischen Geräten und Methoden, um die Erfassung, Kommunikation und Verarbeitung der Sensordaten effizient zu

ermöglichen. Da es theoretisch keinerlei Einschränkungen hinsichtlich der dafür nutzbaren Techniken gibt, werden nachfolgend nur die wichtigsten, mit Hinblick auf die später zu entwerfenden Konzepte, kurz vorgestellt.

Übertragungstechnik

Zuerst wollen wir in diesem Kontext näher auf die Möglichkeiten der Datenübertragung der Mobilgeräte eingehen, die sehr heterogen gestaltet sein kann. So sind neben der konventionellen Datenübertragung über das Mobilfunknetz (z.B. GPRS oder UMTS), über die Mobilgeräte Datenpakete an eine IP-Adresse im Internet verschicken können, ebenso Datenübertragungen durch mobile Ad-hoc-Netzwerke (MANETs) oder eine Mischung aus beiden Techniken denkbar. Während die erwähnten Mobilfunktechniken weitestgehend standardisiert und weit verbreitet sind, ist die Kommunikation über ein Ad-hoc-Netzwerk noch immer eine Randerscheinung. Daher wollen wir diese Technik im Folgenden näher beleuchten.

Eine mobile Ad-hoc-Übertragung ermöglicht eine drahtlose Kommunikation zwischen Mobilgeräten, bei der keine Infrastrukturelemente notwendig sind (siehe zum Beispiel [13]). Der Datenverkehr erfolgt dabei über die Geräte selbst, ohne dass die Vermittlung einer Basisstation notwendig ist. Aufgrund der ständigen Bewegungen der mobilen Geräte kann kein statisches Netzwerk im herkömmlichen Sinn aufgebaut werden, wodurch zusätzliche Herausforderungen entstehen, die mit Techniken einer statischen Netzwerkumgebung nicht gelöst werden können. So tritt zum Beispiel beim Routing von Paketen das Problem auf, dass sich eine Route von einem Knoten A zu einem Knoten B ändern kann, weil sich ein Zwischenknoten außerhalb des Empfangsbereichs der anderen Knoten bewegt hat. Auch wenn MANETs ein noch relativ junges Forschungsgebiet sind, gibt es mittlerweile eine breite Auswahl an Forschungsergebnissen für derartige Probleme, wie zum Beispiel das *Dynamic Source Routing* [25] oder das *Location Aided Routing* [29], welche speziell das angesprochene Routing-Problem lösen. Beim Location Aided Routing verwerten die Knoten Informationen über ihre derzeitige geografische Position, um Entscheidungen beim Routing von Paketen zu treffen. Eine Anwendung dieser Technik findet sich zum Beispiel bei der Durchführung eines Geocasts. Dies ist ein Multicast, der die Eigenschaft besitzt, nur an diejenigen Geräte eine Nachricht zu versenden, welche sich innerhalb eines zuvor definierten geografischen Gebiets befinden.

Wird die Ad-hoc-Kommunikation über eine WLAN-Schnittstelle realisiert, wie es in der Regel der Fall ist, so kann die Übertragung eine Datenrate von bis zu 54 Mbit/s erreichen. Der entscheidende Nachteil im Vergleich zur Mobilfunktechnik liegt jedoch in der geringen Reichweite. Eine typische WLAN-Verbindung erzielt bei guten Bedingungen eine Reichweite von bis zu 100 Metern, unter der Voraussetzung, dass direkter Sichtkontakt zwi-

schen den beiden Kommunikationseinheiten besteht. Diese Reichweite kann jedoch durch Hindernisse, wie Gebäude oder Störsignale entscheidend verkürzt werden.

Für das im Rahmen dieser Masterarbeit zu entwerfende System, werden sowohl die Mobilfunk- als auch die Ad-hoc-Übertragung genutzt. Mobilgeräte übertragen und empfangen Informationen von einem zentralen Server mittels Mobilfunktechnik, während die Kommunikation der Mobilgeräte untereinander über deren Ad-hoc-Schnittstelle abläuft.

Eigenschaften der Mobilgeräte

Die Mobilgeräte, auf denen sich die Sensoren befinden, können allgemein in zwei Klassen unterschieden werden. Zum einen gibt es Sensoren, welche in Fahrzeugen angebracht sind (*vehicular*), und zum anderen Sensoren, die sich auf handgeführten Geräten befinden (*handheld*) [1]. Da für die spätere Umsetzung vorerst nur Daten von Mobiltelefonen ausgewertet werden, beschränkt sich die folgende Erläuterung auf handgeführte Geräte und deren technischen Eigenschaften. Diese können sich jedoch innerhalb eines Fahrzeugs befinden und daher ähnliche Eigenschaften bezüglich ihrer Bewegungsabläufe annehmen, wie die Geräte der ersten Kategorie.

Moderne Mobiltelefone der neuesten Generation (z.B. das Apple iPhone 4 oder das Google Nexus One) unterstützen neben der konventionellen Sprach- und Nachrichtenübertragung eine Vielzahl von Anwendungen, die von der Realisierung eines Navigationssystems über die Musikwiedergabe und den Internetbrowser bis hin zur Bildverarbeitung reichen. Wesentlicher Motor für die Vielfalt der heutzutage erhältlichen Anwendungen ist die für die meisten Systeme leicht erlernbare Programmierschnittstelle (API[1]) zum Betriebssystem des Mobiltelefons. Diese ermöglicht Programmierern, ohne systemspezifische Vorkenntnisse, relativ einfach Programme zu erstellen und zu veröffentlichen. Zudem ermöglicht sie in der Regel ein relativ einfaches Auslesen der Gerätesensoren, so dass die Implementierung einer Urban Sensing Anwendung ohne gerätespezifische Vorkenntnisse leicht möglich ist.

Die Vielfalt dieser Sensoren ist mittlerweile recht groß. Standardmäßig verfügt jedes Mobiltelefon über ein eingebautes Mikrofon, welches für die Aufzeichnung der Stimme beim Telefonat nötig ist. Neben diesem gehören seit einigen Jahren auch Digitalkameras zur Standardausrüstung, welche zu Beginn noch eine relativ niedrige Auflösung hatten, mittlerweile aber mehrere Megapixel erreichen. Während früher nur einige wenige Geräte, überwiegend im Outdoorbereich, über Sensoren wie einen GPS-Empfänger oder einen Kompass verfügten, sind diese seit kurzem in der Mehrzahl der neuen Mobiltelefone ebenfalls enthalten. Diese machen es möglich, ortsbezogene Anwendungen, wie Navigationssysteme, auf den Geräten zu implementieren.

[1] Application Programming Interface

Im Rahmen dieser Arbeit wird lediglich ein GPS-Empfänger als Sensor auf den Mobilgeräten benötigt, sowie die bereits im vorangehenden Abschnitt erwähnten Datenübertragungstechniken. Weitere Details zur Positionsbestimmung via GPS finden sich in Abschnitt 2.3.3.

2.2.4 Anwendungsbeispiele

Um ein besseres Verständnis für die Möglichkeiten und Anwendungen von Urban Sensing zu bekommen, sind nachfolgend beispielhaft Projekte aufgeführt, die einen kurzen Einblick darüber geben, was zurzeit im Bereich Urban Sensing entwickelt wird.

CarTel Das CarTel-System bietet seinen Benutzern Informationen über aktuelle Verkehrsentwicklungen in Echtzeit an. Dazu sammelt es Sensordaten von mobilen Einheiten der Benutzer, wie sie zum Beispiel von der Autoperipherie oder von Smartphones bereitgestellt werden, und leitet diese an einen zentralen Server weiter. Dort werden die gesammelten Daten analysiert, aufbereitet und den Benutzern wiederum in Form von Verkehrsinformationen, wie zum Beispiel als Informationen über Stauvermeidung oder den Zustand des Straßenbelags, bereitgestellt.[36]

CitySense CitySense ist eine Anwendung für Smartphones, welche seinen Benutzern Informationen darüber zur Verfügung stellt, wo in San Francisco gerade die Hotspots des Nachtlebens zu finden sind. Dabei analysiert und aggregiert CitySense anonymisierte Positionsinformationen von Benutzern in Echtzeit und wertet diese aus, um den Benutzern wiederum Informationen über das aktuelle Menschenaufkommen in der Stadt bereitzustellen. Mögliche Informationen sind zum Beispiel, wie viele Benutzer gerade wo unterwegs sind oder welches die meist besuchten Bars, Clubs oder Restaurants der Stadt sind.[12]

CamMobSens CamMobSens ist ein an der University of Cambridge entwickeltes System, welches die Überwachung von Merkmalen der Luftverschmutzung, wie zum Beispiel der Anteil des Kohlenmonoxids, zur Aufgabe hat. Die hierfür benötigten Daten werden von Fußgängern erfasst, welche einen entsprechenden Sensor, der die Luft analysiert, und ein Mobiltelefon, das diese Daten an eine zentrale Datenbank überträgt, mit sich tragen. Diese werden zur Auswertung entsprechend aggregiert und können anschließend auf einer Karte visualisiert werden.[43]

Neben diesen bereits laufenden Projekten sind außerdem noch eine Vielzahl anderer Szenarios möglich. So sind Anwendungen im Gesundheitswesen denkbar, welche ausgewertete Sensordaten nutzen, um Einflüsse auf den Gesundheitszustandes der Teilnehmer, hinsichtlich Umweltfaktoren wie Lärmbelästigung und Temperaturschwankungen, zu ermitteln.

Diese könnten an verschiedenen Orten in der Stadt gesammelt werden und so die aufwändige Platzierung von statischen Sensoren überflüssig machen [10].

Nicht zu vergessen sind die Möglichkeiten, die sich im Bereich der Kartografie ergeben. Partizipatorische Ansätze könnten zum Bespiel die Erstellung einer kommentierten Stadtkarte ermöglichen, welche speziellen Interessengruppen ausgewählte Informationen bereitstellen könnte. So wäre es denkbar, dass ein begeisterter Läufer eine besonders attraktive Joggingroute aufzeichnet und diese Daten anderen interessierten Läufern zur Verfügung stellt. Zudem haben Projekte wie OpenStreetMap gezeigt, dass sich mit einem partizipatorischen Ansatz erfolgreich Karten erstellen lassen, welche ansonsten nur über kommerzielle Anbieter verfügbar sind. Dazu im folgenden Abschnitt mehr.

2.3 Collaborative Mapping

Bis vor wenigen Jahren war die Erstellung von Karten eine Disziplin, die nur von wenigen Fachmännern ausgeübt werden konnte, da die dafür benötige Technologie und das erforderliche Fachwissen nur schwer zugänglich waren. Mit der zunehmenden Verbreitung neuer und billiger Positionierungsverfahren, wie zum Beispiel GPS, und der Vervielfältigung der Kommunikationsmöglichkeiten über Medien wie das Internet oder die Mobilfunktechnik, entstand in den letzten Jahren ein neues Paradigma, welches unter dem Namen Collaborative Mapping bekannt wurde und einen neuartigen Prozess zur Erstellung von Kartenmaterial beinhaltet. Dieser ermöglicht es nun auch Laien auf dem Fachgebiet der Kartografie, durch Nutzung dieser neuen Technologien, gemeinschaftlich Karten zu erzeugen.

Die folgenden Abschnitte haben zum Ziel, die wichtigsten Grundlagen und Techniken des Collaborative Mapping zu erläutern. Dazu wird in Abschnitt 2.3.1 zuerst ein allgemeiner Überblick über die Thematik gegeben, bevor in Abschnitt 2.3.2 erklärt wird, welche Auswirkungen diese neue Art der Kartenerstellung auf die konventionelle Kartografie hat. Anschließend wird in Abschnitt 2.3.3 auf das wichtigste Werkzeug des Collaborative Mapping näher eingegangen - das Global Positioning System (GPS). Abschließend werden in Abschnitt 2.3.4 Beispiele vorgestellt, anhand welcher der Leser einen besseren Eindruck von konkreten Umsetzungen dieses Konzepts gewinnen kann.

2.3.1 Grundlagen

Unter Collaborative Mapping versteht man die gemeinschaftliche Erstellung von Karten durch eine Vielzahl von Teilnehmern, welche geografisches Wissen zusammentragen, das auf einer gemeinsamen Plattform zu einem Gesamtmodell integriert wird. Dieser Ansatz reicht von der kompletten kartografischen Erfassung eines bisher nicht kartografierten Be-

reichs, bis zur interessenspezifischen Annotation bereits vorhandener Karten. Im ersten Fall kartografieren die Teilnehmer neue Straßen und Objekte durch das Aufzeichnen von Koordinaten, welche sie später ins Gesamtmodell übertragen und mit der ihnen bekannten Ortsbezeichnung benennen. Ein Beispiel für diese Art des Collaborative Mapping, auch *User-Generated Mapping* genannt, ist das OpenStreetMap-Projekt [38].

Im Gegensatz zur kompletten Neuerstellung von Karten, haben Projekte wie *Tagzania* [46] oder *Urban Tapestries* [44] lediglich die Anreicherung bereits existierender Karten zum Ziel. So ändern die Teilnehmer nichts an der räumlichen Anordnung der Kartenobjekte, sondern fügen lediglich interessenspezifische Informationen an der entsprechenden Stelle auf der Karte ein. Diese Informationen werden als *Overlay* bezeichnet, da über die bereits vorhandene Karte lediglich eine zusätzliche Informationsschicht gelegt wird. Ein Beispiel für ein Overlay ist die Markierung von beliebten Restaurants oder Laufstrecken in einer Stadt.

2.3.2 Auswirkungen für die Kartographie

Die durch das Collaborative Mapping gesammelten Informationen sind in den meisten Fällen für Dritte kostenlos abrufbar und weiterverwertbar (so genannte *Open Maps*). Damit stehen diese in direkter Konkurrenz zu Anbietern von proprietärem Kartenmaterial, welche ihrerseits Karten anbieten, den Zugang zu diesen aus wirtschaftlichen Gründen aber oft starken Einschränkungen unterwerfen. Grundsätzlich fällt die Erstellung von Karten in den meisten Ländern in den Verantwortungsbereichs des Staats. Diese Karten werden durch speziell ausgebildete Kartografen erstellt, welche an Universitäten über mehrere Jahre in speziellen Studiengängen ausgebildet werden. Die Verfügbarkeit dieser Karten ist dabei von Region zu Region unterschiedlich. Während die USA ihre Geodaten ihren Bürgern frei zur Verfügung stellen, sind diese Daten in Europa und weiten Teilen der Welt nur gegen eine Gebühr, welche zur Finanzierung der Kartografie benötigt wird, erhältlich [20]. Kommerzielle Anbieter lizenzieren diese Daten oder kaufen sie bei größeren Anbietern ein, welche finanziell in der Lage sind eigene Geodaten zu pflegen. Durch dieses System sind die erworbenen Daten mit Kosten verbunden und müssen durch ein Copyright geschützt werden, womit sie nur eingeschränkt genutzt werden können.

Collaborative Mapping umgeht diese Problematik und bietet zusätzlich eine Reihe von Vorteilen gegenüber dem klassischen Ansatz der Kartografie. So liegt der größte Vorzug in den vergleichbar geringen Kosten, welche sich meist auf die Instandhaltung der nötigen Infrastruktur beschränken, da die Geodaten in der Regel von Freiwilligen gesammelt werden und daher keine teuren Spezialisten nötig sind. Ein weiterer Vorteil liegt in der Aktualität der Daten, welche bei einer ausreichend hohen Benutzerzahl überdurchschnittlich hoch ist. Auf der anderen Seite sind derartige Projekte jedoch auch mit großen Schwierigkei-

ten konfrontiert, für die in der Zukunft noch Lösungen gefunden werden müssen. So hat speziell OSM damit zu kämpfen, dass nur ein geringer Bruchteil seiner Nutzer aktiv am Kartografieren mitwirkt und immer noch große Teile der Weltkarte unerschlossen sind [22]. Die Tatsache, dass nahezu jeder Geodaten zum Projekt beisteuern kann, macht zudem die Qualität und Genauigkeit dieser nur schwer überprüfbar. Der Benutzer sollte sich bei deren Verwendung daher immer bewusst sein, dass diese meist ungenauer sind als Geodaten, die aus Methoden der herkömmlichen Kartografie hervorgehen.

Für die Umsetzung eines Collaborative Mapping Systems ist eine Positionierungsmethode notwendig, die jedem Teilnehmer möglichst einfach und genau dessen aktuelle Position mitteilen kann. Dabei ist GPS das mittlerweile wichtigste Hilfsmittel, um aktiv Geodaten zur Kartenerstellung beizusteuern. Waren GPS-Empfänger bei Markteinführung noch groß und teuer, passen sie mittlerweile in jedes handelsübliches Mobiltelefon. Dieser Fortschritt verbindet Collaborative Mapping mit dem Urban Sensing Ansatz, da die Teilnehmer dadurch ihre GPS-Positionsdaten direkt dem System zur Verfügung stellen können. Daher wollen wir uns im nächsten Abschnitt diese Technik im Detail anschauen.

2.3.3 Das Global Positioning System

Allgemeines

Die wichtigste technische Errungenschaft, die Collaborative Mapping erst ermöglicht hat, ist neben der Drahtlos- und Mobilfunktechnik (siehe Abschnitt 2.2.3) das Global Positioning System (GPS) [42]. Mit ihm kann jeder Teilnehmer seine zurückgelegt Route in Form von absoluten Koordinaten festhalten und zur Weiterverarbeitung bereitstellen.

GPS ist ein Satellitennavigationssystem, das aus mehreren Satelliten besteht, welche Signale zur Erde schicken, die ihre jeweilige Position im All beinhalten. Ein GPS-Empfänger wertet diese Signale aus und errechnet aus diesen Koordinaten und den Laufzeiten der Signale seinen aktuellen Standort auf der Erde. Um dabei eine ausreichende Genauigkeit des errechneten Standorts zu gewährleisten, muss der Empfänger die Daten von mindestens vier GPS-Satelliten gleichzeitig empfangen, da er neben den Positionsinformationen der Satelliten auch noch deren genaue GPS-Systemzeit benötigt, um die genaue Entfernung zu den Satelliten zu berechnen. Neben der Bestimmung seines aktuellen Orts kann ein GPS-Empfänger außerdem seine Geschwindigkeit ermitteln, indem er die Ableitung des Ortes nach der Zeit errechnet. Die Grenzen von GPS liegen bei der Ortung innerhalb geschlossener Räume, da diese aufgrund der relativ schwachen Signalstärke des Satellitensignals nicht funktioniert. Außerdem unterliegen GPS-Ortungen gewöhnlich einem Fehler, auf den wir nachfolgend näher eingehen werden.

GPS-Fehlerquellen

Die Genauigkeit einer Positionsbestimmung mittels GPS kann aufgrund zahlreicher Faktoren unterschiedlich hoch sein. Nicht kalkulierbar ist zum Beispiel der gegenwärtige Zustand der Erdatmosphäre, welche durch eine unterschiedliche Beschaffenheit an unterschiedlichen Orten das Satellitensignal verzögern und somit das Messergebnis verfälschen kann [21]. Weitere Faktoren für Ungenauigkeiten können schlechte Winkel zu den momentan sichtbaren Satelliten, Uhrenungenauigkeiten der Satelliten oder der Mehrwegeffekt sein, welcher durch Reflektionen bei der Signalausbreitung auftreten kann. Insgesamt summiert sich die mögliche Ungenauigkeit durch diese Fehlerquellen auf bis zu ± 20 Metern.

Verbesserung bei der Genauigkeit einer GPS-Ortungen können erzielt werden, indem Vergleichsmessungen von stationären Messstationen durchgeführt werden. Durch diese kann der lokale Verfälschungsgrad des Signals ungefähr bestimmt und an die mobilen GPS-Empfänger übermittelt werden, welche daraufhin ihre Ergebnisse entsprechend korrigieren. Diese Techniken werden unter dem Sammelbegriff *Differential Global Positioning System* (DGPS) zusammengefasst und können die Genauigkeit von GPS bis auf wenige Meter verbessern. In den GPS-Empfängern aktueller Mobiltelefone findet dieses Verfahren jedoch noch keine Anwendung.

Um die Unschärfe einer GPS-Ortung abschätzen zu können, liefert jedoch die Mehrzahl der Geräte die Möglichkeit, diese mit Hilfe von so genannten DOP-Werten zu quantifizieren. Details zu diesen Werte, können in Abschnitt 3.1.2 gefunden werden, welcher ein einheitliches Fehlermodell für die Ungenauigkeit von GPS-Ortungen definiert.

Datenformat

Die Mehrheit der aktuellen GPS-Empfänger unterstützen den GPX-Standard[2], welcher im Jahr 2002 veröffentlicht wurde und den Austausch und die Verarbeitung von GPS-Daten um ein vielfaches erleichtert hat [22]. Das GPX-Format speichert einzelne Positionen (Wegpunkte) in absoluten Koordinaten relativ zum WGS84-Bezugssystem, welches jede Position auf der Erde eindeutig durch drei räumliche Koordinaten angibt. Zur Aufzeichnung von Routen unterstützt das Format die Erfassung von *GPS-Tracks*, die aus einer Folge von kurz hintereinander aufgezeichneten Wegpunkten bestehen. Mit Hilfe von Tracks können somit Straßenverläufe modelliert werden oder die zurückgelegte Route von Benutzern gespeichert werden.

[2] Offizielle Webseite: *http://www.topografix.com/gpx.asp*

2.3.4 Anwendungsbeispiele

Während es für partizipatorische Ansätze des Collaborative Mapping bereits mehrere erfolgreiche Umsetzungen gibt, ist die Zahl der Veröffentlichungen und Projekte, die ein opportunistisches Konzept bei der Erfassung der Geodaten verfolgen, recht überschaubar. Mit dem Amsterdam RealTime Project wurde ein erster Ansatz in diese Richtung unternommen, der jedoch mehr als Demonstration fungierte. Nachfolgend wollen wir dieses Projekt kurz vorstellen, bevor wir dann einen Blick auf das wichtigste Projekte werfen, das einen partizipatorischen Ansatz verfolgt — OpenStreetMaps.

Amsterdam RealTime Project

Das Amsterdam RealTime Project [4] ist ein im Jahr 2002 über zwei Monate durchgeführter Versuch der Visualisierung von „mentalen Karten“, welche die Einwohner Amsterdams von ihrer Stadt in ihren Köpfen haben. Dazu wurden die mehr als 60 freiwilligen Teilnehmer mit einem GPS-Empfänger und einem PDA, der via GPRS über eine ständige Verbindung zu einem zentralen Server verfügt, ausgestattet. Der GPS-Empfänger ermittelte in kurzen Abständen seine aktuelle Position, welche in Echtzeit an den Server gesendet wurde, der diese auf einer schwarzen Fläche farbig visualisierte. Oft besuchte Plätze wurden dabei in einer anderen Farbe dargestellt als weniger oft besuchte. Die daraus resultierenden Routen wurde in Echtzeit auf der Aufstellung *Maps of Amsterdam 1866-2000* auf eine Wandtafel projiziert, um den Besuchern der Ausstellung die neuen technischen Möglichkeiten der Kartografie zu demonstrieren. Auf der Wandtafel wurden jedoch lediglich die übertragenen GPS-Routen angezeigt und übereinander gelegt. So entstand mit der Zeit ein Bild, welches einzelne Straßen nicht mehr eindeutig darstellte, da die erfassten Straßen, wegen der Ungenauigkeit der GPS-Ortungen, durch mehrere unabhängige Messungen nicht mehr eindeutig waren.

OpenStreetMaps

Das wohl größte und bekannteste, der mittlerweile existierenden Collaborative Mapping Projekte, ist OpenStreetMaps (OSM)[22]. Ziel von OSM ist es, einen freien Zugang zu geografischen Daten zu bieten und damit unabhängig von Karten proprietärer Anbieter zu werden. Auch wenn scheinbar bereits kostenloses Kartenmaterial durch Anbieter wie *Google* oder *Falk* im Internet existiert, ist der Zugang zu diesem an gewisse Restriktionen und Lizenzen, hinsichtlich der Weiterverwendung der Daten, geknüpft. Diesen Umstand versucht OSM durch eine offene Lizenz zu umgehen, die jegliche Nutzung des Kartenmaterials erlaubt. Die abgeleiteten Werke müssen lediglich die Herkunft der Daten angeben und unter der gleichen Lizenz stehen.

Bei OSM kann die zentrale Weltkarte entweder in Form von Rohdaten oder direkt in Form von geografischen Karten abgerufen werden. Diese Karte ist über einen Onlineeditor zugänglich, über den jeder Benutzer Änderungen an ihr vornehmen kann, um zum Beispiel Straßen zu benennen oder zu korrigieren. Die Kartendaten werden bei OSM gemeinschaftlich durch die registrierten OSM-Benutzer gesammelt, welche über eine Schnittstelle Daten im GPX-Format auf den Server laden können, die sie mit ihrem GPS-Empfänger aufgezeichnet haben. Wer kein GPS-Gerät besitzt, kann sich trotzdem am Kartieren beteiligen, indem er Wege in die Karte selbst einzeichnet. Anschließend können die gesammelten Daten mit den jeweiligen Ortsbezeichnung versehen werden.

In technischer Hinsicht nutzt OSM eine zentrale Datenbank, welche die kartografischen Daten hält. Freiwillige arbeiten an der Instandhaltung und Weiterentwicklung der technischen Infrastruktur, sowie der graphischen Schnittstelle. Die Rohdaten können über ein eigenes XML Format (OSM XML) von der Datenbank extrahiert und von Anwendungen genutzt werden.

Obwohl es bereits eine Reihe von Standards im Bereich der geografischen Modellierung gibt, verfolgt OSM einen Ansatz der Einfachheit und lässt diese aufgrund ihrer Komplexität außer Acht [22]. OSM arbeitet im Wesentlichen nur mit Knoten, welche durch Breiten- und Längengrad geografisch genau bestimmt sind. Durch Verbinden mehrere Knoten entstehen Wege und Flächen. Um komplexere Objekte zu erzeugen, können Elemente zu Relationen gruppiert werden. Jedem Element kann außerdem ein Wertepaar zugeordnet werden, welches die Eigenschaften des Elements beschreibt. So kann eine Straße zum Beispiel als Autobahn oder als Fußgängerweg klassifiziert werden.

2.4 Stand der Forschung

Für die Umsetzung der automatisierten Kartenerstellung sind Techniken aus mehreren Gebieten zu betrachten. Während man beim Vergleich und der Verarbeitung von GPS-Tracks im Bereich der Graphentheorie fündig wird, sind für andere Aufgabenbereiche, wie zum Beispiel bei der Definition geeigneter Qualitätsmetriken, keine allgemeinen Lösungen für die vorliegende Problemstellung bekannt. Daher beschränken wir uns im folgenden Abschnitt auf die Betrachtung von Methoden zur Analyse von GPS-Tracks. Im Anschluss daran wird der wissenschaftliche Beitrag der vorliegenden Masterarbeit noch einmal gesondert hervorgehoben.

2.4.1 Verfahren zur Analyse von GPS-Tracks

Bei der Erfassung von GPS-Tracks ist ein Verfahren nötig, das diese Tracks analysiert und verarbeitet. Dabei hat die inhärente Ungenauigkeit von GPS-Ortungen zur Folge, dass Tracks, die auf der gleichen Straße erfasst wurden, unterschiedliche Wegpunkte enthalten können. Um aus einer Menge von GPS-Tracks ein einheitliches Straßenmodell zu erhalten, gibt es eine Reihe von Verfahren, welche sich mit diesem Problem beschäftigen. Diese sollen nachfolgend vorgestellt werden.

- Der erste Ansatz zur Identifikation mehrerer Tracks, die den gleichen Weg beschreiben, wird in der Arbeit von Morris et al. [37] vorgestellt. Die erfassten Tracks werden zuerst in einen Graphen umgewandelt, welcher aus Knoten (Kreuzungen) und Kanten (Straßenabschnitte) besteht. Durch Anwendung von drei Reduktionsmethoden wird der Graph anschließend so reduziert, dass typische Anzeichen einer doppelten Erfassung erkannt und durch Interpolation der jeweiligen Kanten reduziert werden. Dabei stütze das Verfahren auf feste Grenzwerte bei der Analyse von Ähnlichkeiten von Kanten und vernachlässigt somit die probabilistische Komponente der einzelnen Wegpunkte.

- In eine ähnliche Richtung geht die Arbeit von Chen at al. [11]. Sie beschreibt einen Algorithmus der aus einer Menge von Wegpfaden ein Straßennetz generiert. Dieser versucht dabei im Wesentlichen für eine Menge von Pfaden Gemeinsamkeiten herauszufinden, um aus den zusammengehörigen Pfaden dann eine einzige Straße zu generieren. Dabei werden außer den Koordinaten der einzelnen Pfade keine weiteren Informationen berücksichtigt.

- Ziel der Arbeit von Davies et al. [15] ist es, aus einer Menge von GPS-Tracks, welche von Fahrzeugen erfasst wurden, einen Graphen zu erzeugen, dessen Kanten die Straßen und dessen Knoten Kreuzungen bilden. Sie stellt dabei einen Algorithmus vor, welcher zunächst den zu erfassenden Bereich in ein Netz von Gitterzellen mit einer Seitenlänge von wenigen Metern unterteilt. Für jede dieser Zellen wird nun gezählt, wie viele Wegpunkte sich innerhalb der Zelle befinden. Zusätzlich wird jede Zelle, die sich auf direktem Weg zwischen zwei aufeinanderfolgenden Wegpunkten in einem Track befindet, ebenfalls berücksichtigt. Nachdem ein Filter die Unschärfe, die in der Verteilung der Werte auf die einzelnen Zellen entsteht, ausgeglichen hat, wird anhand eines Grenzwertes binär entschieden, welche Zellen Teil einer Straße sind. Mit Hilfe eines Voronoi-Diagramms wird anschließend für jede Straße eine zentrale Linie bestimmt, welche sich aus der Menge der zugehörigen Gitterzelle ergibt.

- Bruentrup et al. [7] stellen ein Verfahren vor, welches erfasste GPS-Tracks verarbeitet und dann in ein Gesamtmodell integriert. Das Verfahren gliedert sich in drei Schritte.

Zuerst werden für einen zu verarbeitenden Track Wegpunkte bereits erfasster Straßen gesucht, welche sich in der Nähe des Tracks befinden. Danach wird für jede in Frage kommende Straße berechnet, welche am ehesten einen ähnlichen Verlauf wie der zu verarbeitende Track hat. Im letzten Schritt werden aus diese Straße und dem übergebenen Track der Durchschnitt berechnet und im Straßenmodell gespeichert. Findet sich keine passende Straße, welche sich in der Nähe des eingehenden Tracks befindet, wird der Track als neue Straße aufgenommen.

- Die Arbeit von Roth [41] beschäftigt sich mit der gleichen Problematik, enthält jedoch einen Ansatz, der die Ungenauigkeit einzelner GPS-Wegpunkte berücksichtigt. Dazu vergleicht er die aufgezeichneten Tracks paarweise und überprüft diese, mit einem auf Histogrammen basierenden Verfahren, auf Ähnlichkeit. Er betrachtet die GPS-Wegpunkte dabei nicht als absolute, sondern als probabilistische Werte, deren Genauigkeit über deren Varianz bestimmt sind. Wird für zwei Tracks eine Ähnlichkeit festgestellt, so werden deren Wegpunkte und Varianzen zu einem neuen Track interpoliert.

Ziel der vorgestellten Verfahren ist die automatische Generierung einer Straßenkarte aus einer Menge von GPS-Tracks. Keine dieser Ansätze betrachtet jedoch die Möglichkeit zur gezielten Steuerung der Erfassung der GPS-Tracks und der damit verbundenen Chance Ressourcen einzusparen, wie es in dieser Arbeit umgesetzt werden soll. Um unser Ziel zu erreichen ist ein Verfahren nötig, um die GPS-Tracks zu einer eindeutigen Straßenkarte zu integrieren. Da das Verfahren von Roth mehr Informationen bei der Verarbeitung berücksichtigt als die anderen, werden wir bei der Umsetzung hierauf aufbauen. Eine genauere Erläuterung dieses Verfahrens kann daher in Abschnitt 3.2.2 gefunden werden. Da im späteren Verlauf noch andere wissenschaftliche Vorarbeiten referenziert werden, wie zum Beispiel im Bereich des Lokationsmanagements in Abschnitt 3.4.2, diese jedoch jeweils nur zu einem kleinen Teil genutzt werden, sind deren Referenzen in den jeweiligen Abschnitt im Konzeptteil zu finden.

2.4.2 Beitrag dieser Arbeit

Der wissenschaftliche Beitrag, den diese Arbeit leistet, liegt darin ein Konzept zu entwerfen, das eine automatische Erstellung von Straßenkarten durch Mobilgeräte ermöglicht. Da die Ressourcen von Mobilgeräten begrenzt sind, muss dieses Konzept so gestaltet werden, dass die Ressourcenschonung der Mobilgeräte im Vordergrund steht.

Dieses Konzept nutzt Ergebnisse aus dem zuvor vorgestellten Abschnitt und erweitert sie so, dass bei der automatisierten Erstellung von Straßenkarte zum einen die Ressourcen

der Mobilgeräte möglichst geschont werden, und zum anderen durch Integration der GPS-Daten eine möglichst genaue Straßenkarte entsteht.

Um dies zu erreichen, muss die Analyse und Verarbeitung der GPS-Tracks so erweitert werden, dass daraus sinnvolle Qualitätsaussagen für die resultierenden Straßen ableitbar sind. Mithilfe dieser Qualitätsdaten kann ein adaptiver Erfassungsalgorithmus entwickelt werden, der dynamisch Entscheidungen darüber trifft, welche Bereiche der zu erstellenden Karte von den Mobilgeräten gezielt zu erfassen sind. Die richtige Kombination dieser Techniken ermöglicht ein Verfahren, welches die Ressourcen der teilnehmenden Mobilgeräte schont, ohne einen allzu drastischen Qualitätsverlust bei der Erstellung der Straßenkarte zu erleiden. Die Schonung der Ressourcen spielt beim Urban Sensing eine bedeutende Rolle, da die Benutzer der Mobiltelefone eine derartige Anwendung weniger akzeptieren würden, wenn dadurch die Akkulaufzeit ihrer Mobiltelefone entscheidend verkürzt würde.

Kapitel 3

Konzeptionelle Umsetzung

Nachdem im vorangegangenen Kapitel die theoretischen Grundlagen gelegt und der wissenschaftliche Beitrag dieser Masterarbeit abgesteckt wurde, kann nun die Ausarbeitung der Konzepte beginnen, welche dem zu entwerfenden System zu Grunde liegen.

Dafür wird zuerst in Abschnitt 3.1 ein Systemmodell entworfen, welches alle Bestandteile der Problemstellung in einem Modell abstrahiert. Darauf aufbauend, werden nachfolgend in Abschnitt 3.2 die Algorithmen definiert, mit denen das System arbeitet, um Informationen über Straßen zu erfassen und einheitlich in ein zentrales Modell zu integrieren. Damit das System dynamisch auf qualitative Unterschiede hinsichtlich der Abbildungsgenauigkeit der bisher erfassten Straßen reagieren kann, wird anschließend in Abschnitt 3.3 ein Qualitätsmodell definiert, mit welchem der Server, die zentrale Komponente des Systems, den Erfassungsprozess so anpassen kann, dass die Ressourcen der beteiligten Mobilgeräte möglichst geschont werden. Die Details dieser dynamischen Anpassung werden in Abschnitt 3.4 festgelegt. Darüber hinaus beinhaltet dieser Abschnitt zusätzliche Überlegungen, welche die Klassifizierung erkannter Straßen und allgemeine Optimierungspotentiale umfassen.

3.1 Systemmodell

Um ein implementierfähiges Konzept erstellen zu können, muss die Aufgabenstellung in einem Modell abgebildet werden, welches alle wichtigen Eigenschaften des Systems und dessen Umwelt beinhaltet. Dafür ist es notwendig, dass wir zuerst in Abschnitt 3.1.1 den vorliegenden Ist-Zustand der Problemstellung in Form eines Weltmodells erfassen, welches die Komplexität der realen Welt soweit wie nötig abstrahiert. Da GPS-Sensordaten eine wesentliche Rolle bei der Umsetzung des Systems spielen, muss eine einheitliche Definition für den Fehler, der bei der GPS-Ortung eintritt, festgelegt werden. Dies geschieht in Ab-

schnitt 3.1.2. Abschließend werden wir in Abschnitt 3.1.3 auf die, für das Konzept nötigen, technischen Geräte eingehen und die Anforderungen an diese Geräte möglichst abstrakt definieren, um den Konzeptteil auf einer wohl definierten Plattform aufzubauen.

3.1.1 Weltmodell

Als Grundlage für die konzeptionellen Entwürfe ist ein Modell nötig, welches alle systemrelevanten Objekte der realen Welt möglichst einfach und originalgetreu abstrahiert. Diese Objekte sind zum einen der zu kartografierende geografische Bereich mit den sich darin befindlichen Straßen und zum anderen die mobilen Sensoren, welche sich auf diesen Straßen fortbewegen, indem sie von Personen mitgeführt werden.

Um die geografische Position dieser Objekte zu beschreiben, nutzen wir Koordinaten bezüglich des WGS84-Bezugssystems, welches ebenfalls von GPS genutzt wird, um so unnötige Umrechnungen von Koordinaten verschiedener Bezugssysteme zu vermeiden. Das WGS84-Bezugssystem beschreibt die Erde annähernd durch einen Ellipsoid. Da es jedoch unser Ziel ist, eine zweidimensionale Karte zu erstellen, muss dieses räumliche Objekt auf eine Ebene projiziert werden. Dazu vernachlässigen wir einfach die Höhenkoordinate, da der zu erfassende geografische Bereich (Zielbereich) auf dem Ellipsoid eine vergleichbar kleine Fläche einnimmt. Dadurch wird die Krümmung des Ellipsoids ignoriert und der Zielbereich kann über die Eckpunkte einer zweidimensionalen Fläche beschrieben werden (siehe Abbildung 3.1).

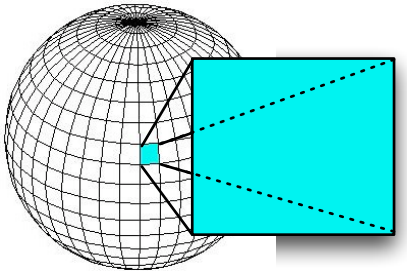


Abbildung 3.1: Abbildung vom WGS84-Bezugsystem auf 2D-Fläche

Nun gilt es, Objekte der realen Welt im Weltmodell möglichst geeignet zu abstrahieren. Eine im Zielbereich vorhandene Straße bilden wir im Weltmodell durch eine Sequenz von Punkten ab, welche durch absolute Koordinaten beschrieben sind. Zwei aufeinanderfolgende Punkte einer Straße sind durch eine Kante verbunden, so dass für jede Straße ein Linienzug entsteht, der diese so exakt wie möglich beschreibt.

Das Ziel der Kartengenerierung besteht nun darin, Verlauf und Position dieser Straßen

bestmöglich mit Hilfe von mobilen Sensoren, welche sich über diese Straßen fortbewegen, in einer Datenstruktur zentral zu erfassen. Jeder mobile Sensor hat dabei zu jeder Zeit eine eindeutige Position, die er mit Hilfe einer GPS-Ortung bestimmen kann, wobei diese Positionsbestimmung immer einer Ungenauigkeit unterliegt, die das Ergebnis verfälscht. Diese Ungenauigkeit wollen wir im nächsten Abschnitt einheitlich definieren.

3.1.2 GPS-Fehlermodell

Wie Abschnitt 2.3.3 bereits erwähnt, beinhaltet jede GPS-Ortung eine inhärente Ungenauigkeit, welche eine absolut zuverlässige Positionsbestimmung unmöglich macht. Während einige Fehlerquellen, wie zum Beispiel der Zustand der Erdatmosphäre, von einem einfachen GPS-Empfänger nicht ermittelt werden können, kann jedoch fast jeder Empfänger, durch die Konstellation der für ihn sichtbaren Satelliten, seine Messgenauigkeit hinsichtlich dieser Satellitengeometrie ermitteln. Diese wird in so genannten DOP-Werten (Dilution of Precision) erfasst und mit jeder Positionsbestimmung gespeichert. Daher nehmen wir im Folgenden an, dass für jede GPS-Ortung, neben den Koordinaten, der Zeit und der Geschwindigkeit, auch die DOP-Werte gespeichert werden.

Unter der Annahme, dass mehrere von einem GPS-Empfänger ermittelte Positionen normalverteilt um dessen tatsächliche Position sind, lässt sich mit Hilfe des HDOP-Werts (Horizontal DOP) und dem gerätespezifischen σ_{UERE} (User Equivalent Range Error) die Standardabweichung σ der Positionsbestimmung wie folgt bestimmen [32]:

$$\sigma = HDOP \cdot \sigma_{UERE}$$

Wir nehmen an, dass für jeden GPS-Empfänger der Wert σ_{UERE} verfügbar und der HDOP-Wert aus einer GPS-Ortung ableitbar ist, so dass σ für jede GPS-Ortung bestimmt werden kann.

Die resultierende Standardabweichung σ entspricht im Allgemeinen einem Konfidenzintervall der Größe $c = 68.27\%$ [40]. Das bedeutet anschaulich, dass die Abweichung von der Position, welche durch die GPS-Ortung bestimmt wurde, zur tatsächlichen Position des Empfängers mit einer Wahrscheinlichkeit von 68.27% nicht größer als σ ist (siehe Abbildung 3.2).

Durch die Definition von σ für jede einzelne GPS-Positionsbestimmung erhalten wir ein Fehlermodell, mit dem wir Ungenauigkeiten bei der Verarbeitung der GPS-Daten einheitlich berücksichtigen können. Zusätzlich erhalten wir mit Hilfe der allgemeinen Beziehung $var = \sigma^2$ für jede Messung die Varianz, welche wir in späteren Abschnitten verwenden werden. Bevor wir uns den eigentlich Konzepten zuwenden können, müssen wir noch eine geeignete Abstraktion für die im System verwendeten Hard- und Softwarekomponenten

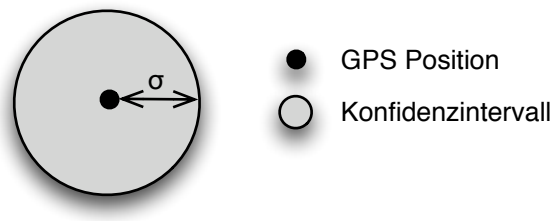


Abbildung 3.2: Konfidenzintervall für σ

finden.

3.1.3 Hardwarekomponenten

Wie bereits erwähnt, sind für die Erfassung der Straßenverläufe Mobilgeräte, welche Straßenpositionen über GPS-Empfänger erfassen, und ein Server, welcher diese Daten verarbeitet und die Erfassung koordiniert, nötig. Zur Einfachheit wollen wir die Anforderungen an diese Geräte nachfolgend grob abstrahieren, ohne ein allzu spezielles Systemmodell zu erhalten.

Mobilgeräte

Wir setzen voraus, dass jedes Mobilgeräte zum einen über ein Mobilfunknetz und zum anderen via Ah-hoc-Technik kommunizieren kann. Als Übertragungstechnik für die Ad-hoc-Kommunikation verfügen die Geräte über ein WLAN-Interface, welches über eine Reichweite von circa 100 Meter verfügt. Ein Gerät kann die zurückgelegte Route einer Person erfassen, indem es GPS-Positionsbestimmungen durchführt und diese Daten dann speichert. Dabei kann es für jede Position, neben den Koordinaten, auch die aktuelle Geschwindigkeit des Teilnehmers und die Varianz der Messung, welche mit Hilfe der DOP-Werte ermittelt wird, speichern.

Server

Die mobilen Geräte senden ihre erfassten Daten zur Weiterverarbeitung an einen zentralen Server. Dieser besitzt eine Kommunikationsschnittstelle und kann so mit jedem ihm bekannten Mobilgerät gezielt kommunizieren. Des Weiteren verfügt er über eine Datenbank, welche alle erfassten Straßen speichern kann. Über eine entsprechende Schnittstelle steht er außerdem für externe Anfragen, zur kartografischen Erfassung eines geografischen Bereichs,

bereit.

3.2 Straßenerfassung

Ziel der Straßenerfassung ist es, mit Hilfe der GPS-Tracks, welche die Mobilgeräte an den Server übertragen, ein möglichst genaues Abbild der Straßen des Weltmodells auf dem Server zu erstellen. Dazu wird zuerst in Abschnitt 3.2.1 definiert, wie das System die erfassten Daten verwaltet und speichert. Danach wird die eigentliche Erfassung der GPS-Tracks vorgestellt, welche sich speziell mit der Problematik, die bei der Integration neu erfasster Tracks entsteht, befasst. Wenn der Server einen GPS-Track von einem Mobilgerät empfängt, muss er zuerst feststellen, ob sich das darin beschriebene Straßenstück bereits im Straßenmodell befindet und wie er dieses entsprechend integrieren kann. Eine dafür geeignete Technik wird in Abschnitt 3.2.2 vorgestellt.

3.2.1 Datenmodelle und -strukturen

Mit dem in Abschnitt 3.1.1 definierten Weltmodell ist eine Abstraktion der Straßen gegeben, welche durch das System erfasst werden sollen. Um die Straßen des Weltmodells in unserem System repräsentieren zu können, sind Datenstrukturen nötig, welche die erfassten Straßeninformationen möglichst effizient speichern. Bevor wir jedoch auf die Speicherung dieser Straßen eingehen, treffen wir Konventionen darüber, wie der zu erfassende Zielbereich im System hinterlegt wird.

Zielbereich

Eine Anfrage an das System beginnt mit der Angabe der Koordinaten der Eckpunkte des Zielbereichs, der zu kartografieren ist. Wir nehmen an, dass dieser Bereich auf einer projizierten Karte der Erdoberfläche eine rechteckige Form hat, so dass dieser durch die Koordinaten von zwei Eckpunkten eindeutig beschrieben werden kann (siehe Abschnitt 3.1.1). Hat der geforderte Zielbereich eine andere Form, so wird das minimale Rechteck, welches den Zielbereich vollständig einschließt, berechnet und verwendet. Dieses Rechteck wird im System durch die Speicherung der absoluten Koordinaten der Eckpunkte hinterlegt.

Wie im späteren Verlauf dieser Arbeit an mehreren Stellen deutlich wird, ist es sinnvoll den Zielbereich weiter zu unterteilen. Dazu definieren wir ein Netz von quadratischen Gitterzellen mit der Seitenlänge g (siehe Abbildung 3.3). Ausgehend von der linken oberen Ecke[1] des Zielbereichs, werden diese Zellen mit $C_{x,y}$ adressiert, wobei $C_{1,1}$ die linke obere

[1] Die linke obere Ecke sei immer die Ecke, welche am nächsten zur Nord-West-Richtung liegt.

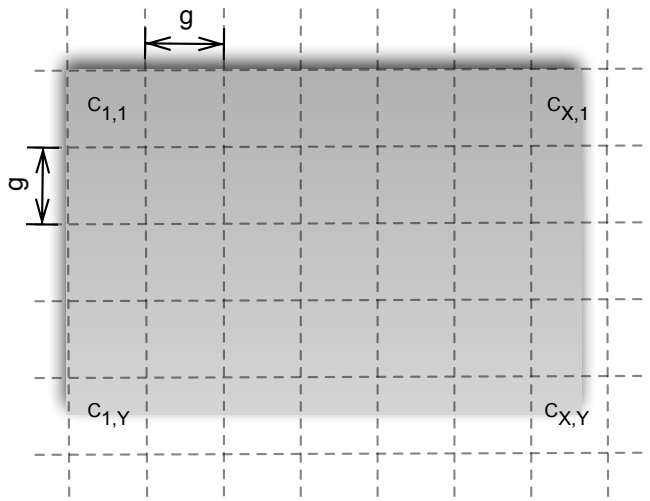


Abbildung 3.3: Zielbereich mit Gitterzellen der Seitenlänge g

Zelle und $C_{X,Y}$ die rechte untere Zelle ist. X und Y sind so gewählt, dass das Gitternetz den Zielbereich vollständig einschließt. Um im weiteren Verlauf Statusinformationen über Gitterzellen auszutauschen, sendet der Server bei Bedarf einen Vektor der Länge $X \cdot Y$ an die Mobilgeräte, wobei jede Stelle diese Vektors einer Gitterzelle entspricht. Welchen Status eine Gitterzelle annehmen kann, zeigt der später folgende Abschnitt 3.4.3.

Straßenmodell

Straßenverläufe werden durch GPS-Empfänger, die sich auf den Mobilgeräten befinden, erfasst und in Form von GPS-Tracks gespeichert. Diese werden in regelmäßigen Abständen an den Server übermittelt, der diese verarbeitet und intern, im so genannten Straßenmodell, speichert. Es werden daher im Folgenden grundsätzlich zwei Arten von Positionsdaten unterschieden:

- GPS-Tracks sind Rohdaten (Primärdaten), welche von den mobilen Sensoren erfasst, an den Server übermittelt und dort verarbeitet werden, bevor sie Änderungen im Straßenmodell bewirken können.
- Straßen gehen aus der Verarbeitung der GPS-Tracks hervor und spiegeln den bisher erfassten Zustand des Zielbereichs wider (Sekundärdaten). Durch die Verarbeitung von Tracks können vorhandene Straßen verändert werden oder neue Straßen entstehen (siehe Abschnitt 3.2.2).

Um das Modell möglichst einfach zu halten, werden Straßen analog zur Beschreibung von GPS-Tracks modelliert (siehe Abschnitt 2.3.3) . Das bedeutet, dass eine Straße r aus einer Aneinanderreihung von Wegpunkten $WP_0, WP_1, \cdots, WP_N$ besteht, wobei jeder Punkt eine X- und eine Y-Koordinate besitzt. Wie bereits zuvor erwähnt (siehe Abschnitt 3.1.1), wird auf die Betrachtung der Z-Koordinate verzichtet. Der eigentliche Straßenverlauf wird durch Kanten modelliert, welche sich ergeben wenn für jedes i mit $0 \leq i < N$ zwischen WP_i und WP_{i+1} die kürzeste Verbindung bestimmt wird. Kreuzungen entstehen durch den Schnitt von zwei Straßen, diese werden aber nicht extra modelliert, sondern sind nur bei der Visualisierung der Straßen ersichtlich. Drückt man die Verbindung zwischen zwei benachbarten Punkten als Teilbereich einer Funktion aus, so wären die zugehörigen Funktionen dieses Ansatzes Geraden. Für die spätere Visualisierung der erfassten Straßen, können jedoch auch Funktionen höheren Grades verwendet werden (z.B. Splines [16]), da diese den natürlichen Verlauf von Straßen originalgetreuer darstellen. Den folgenden Algorithmen legen wir jedoch, aus Gründen der Komplexität, eine lineare Funktion zur Verbindung der Wegpunkte zu Grunde.

3.2.2 Erfassung von GPS-Tracks

Das Vorgehen bei der Erfassung der im Weltmodell befindlichen Straßen, besteht aus einer Reihe von Teilaufgaben, die von der Aufzeichnung der Wegpunkte bis hin zur Speicherung dieser im Straßenmodell reicht. Nachstehend werden diese erläutert.

Erfassung der Wegpunkte

Die Erfassung der Wegpunkte erfolgt durch die Mobilgeräte, auf denen eine entsprechende Software installiert ist, welche die Koordination der Sende- und Sensoraktivitäten des Geräts übernimmt. Diese führen jede t_{GPS} Sekunden eine GPS-Ortung durch und verknüpfen diese sequentiell zu einem Track. Dabei gilt, dass je kürzer t_{GPS} gewählt wird, desto genauer wird das Abbild des Straßenverlaufs erfasst. Auf der anderen Seite muss jedoch berücksichtigt werden, dass die Wahl eines kleinen t_{GPS} den Energieverbrauch der Mobilgeräte stark erhöht und damit unerwünscht ist. Optimierungen dieses Werts werden an späterer Stelle vorgestellt (siehe Abschnitt 3.4.10).

Nach einer festgelegten Zeit (siehe Abschnitt 3.4.5) übertragt ein Mobilgerät seinen geraden aufgezeichneten GPS-Track an den Server. Durch die potentiellen Ungenauigkeiten der GPS-Ortungen und der Vielzahl an möglichen Wegpunkten, die eine einzige Straße beschreiben können, muss der GPS-Track, bevor er als Straße im Straßenmodell gespeichert werden kann, einem Vergleichsverfahren unterzogen werden. Dieses untersucht für einen gegebenen Track, ob dieser eine Straße beschreibt, die im Straßenmodell bereits vorhanden

ist. Ist dies der Fall, müssen Track und Straße fusioniert werden, wodurch neue Punkte entstehen, welche die Informationen aus beiden Komponenten integrieren.

Wie bereits erläutert, gibt es mehrere Verfahren, die sich mit der Problematik dieser Fusion beschäftigen. Für den vorliegenden Fall wählen wir das Verfahren von Roth [41], da es durch die Berücksichtigung der Ungenauigkeiten der GPS-Ortungen mehr Informationen verwendet als die anderen Methoden, die in Abschnitt 2.4.1 vorgestellt wurden. Im nachstehenden Abschnitt wird dieses Verfahren schemenhaft erläutert, da die späteren Konzepte dieser Masterarbeit auf diesem aufbauen.

Probleme beim Vergleich von Tracks

Generell kann es durch die Unschärfe bei GPS-Ortungen zu einer Reihe von Problemen kommen, wenn man versucht aus einer Menge von GPS-Tracks eindeutige Straßenverläufe zu generieren. Roth unterscheidet hier drei mögliche Ausprägungen dieser Probleme, welche in Abbildung 3.4 exemplarisch dargestellt sind und an dieser Stelle kurz erläutert werden sollen:

- Das **Line Clustering Problem** tritt durch den möglichen zeitlichen Versatz der GPS-Ortungen auf. Da diese in kurzen, aufeinanderfolgenden Intervallen durchgeführt werden, können die Wegpunkte zweier GPS-Tracks zwar auf der gleichen Straße erfasst worden sein, aber dennoch unterschiedliche Koordinaten haben, da die GPS-Ortungen an verschiedenen Punkten der Straße stattfanden.

- Das **Offset Problem** beschreibt die Situation, die sich ergibt, wenn zwei Tracks auf der gleichen Straße erfasst wurden, aber sich die Koordinaten ihrer Wegpunkte durch einen konstanten Offset unterscheiden. Dahinter steht die Überlegung, das eine GPS-Ortung zwar oft mit einem Fehler behaftet ist, dieser aber über eine kurze Zeitspanne konstant bleibt. So weisen die Koordinaten der Wegpunkte eines GPS-Tracks zwar oft einen Fehler gegenüber der tatsächlichen Position des Mobilgeräts auf, dieser Fehler bleibt aber für weitere Wegpunkte des Tracks konstant. Daher ist bei Tracks, die auf der gleichen Straße zu unterschiedlichen Zeitpunkten erfasst wurden, eine Symmetrie beim Straßenverlauf zu erkennen, auch wenn die Positionen der Wegpunkte durch einen Offset verschoben sind. Liegt diese Situation vor und ist der konstante Offset, der zwischen den Wegpunkten beider Tracks liegt, kleiner als die doppelte GPS-Ungenauigkeit, kann davon ausgegangen werden, dass beide auf der gleichen Straße erfasst wurden. Im Umkehrschluss sollten Tracks, bei denen die Wegpunkte nahe beieinander liegen, deren Offset jedoch nicht konstant ist, nicht so behandelt werden, als wären sie auf der gleichen Straße erfasst worden.

- Das **Segment Problem** beschreibt das Problem, dass ein Track teilweise eine Straße beschreibt, die von einem anderen Track beschrieben wird. Daher gilt es zu bestimmen, welche Wegpunkte diese Tracks gemeinsamen haben und welche nicht.

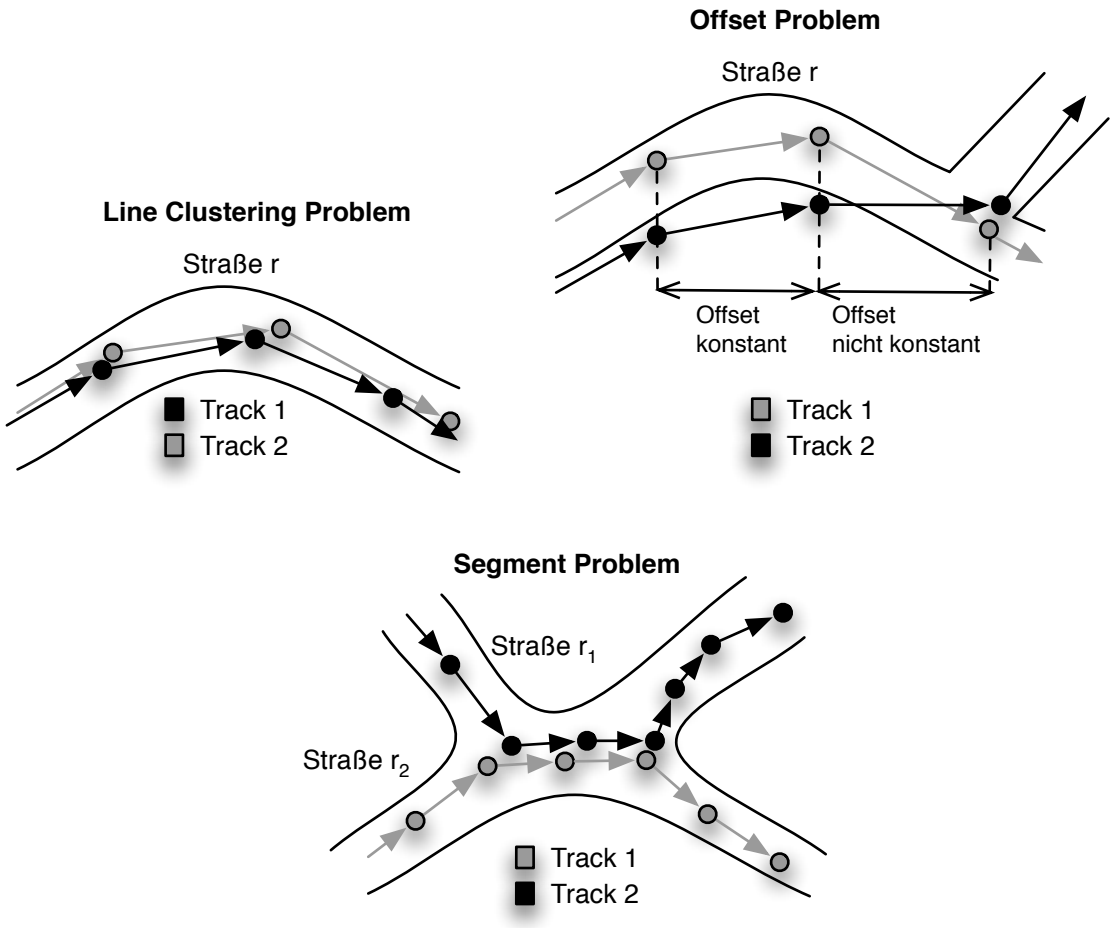


Abbildung 3.4: Mögliche Probleme beim Vergleich von zwei Tracks

Vergleich von Tracks

Roth stellt in seiner Arbeit ein Vergleichsverfahren vor, welches die drei oben beschriebenen Probleme löst. Dieses analysiert für zwei gegebene Tracks, ob ein Teilstück in den Tracks existiert, welches in beiden ähnlich und lediglich durch einen konstanten Offset verschoben ist. Wird ein solches Teilstück gefunden, ist dies ein Zeichen dafür, dass beide Tracks den gleiche Straßenabschnitt beschreiben, jedoch einem unterschiedlichen hohen, aber konstanten, Fehler bei der GPS-Ortung unterlagen.

Dazu werden jeweils zwei Tracks miteinander verglichen. Ausgehend von einem der beiden Tracks, wird von jedem Wegpunkt dieses Tracks eine Verbindung zum anderen Track durch eine Gerade gelegt, welche dort entweder auf einen Wegpunkt oder eine Kante des anderen Tracks trifft. Diese Geraden sind parallel zueinander und liegen im Vergleich zur Senkrechten im einem Winkel von α. Anschließend werden die Längen der Strecken, welche zwischen den Schnittpunkten der Geraden mit den Tracks entstehen, bestimmt und in einem Histogramm visualisiert, welches es nun ermöglicht, mehrere hintereinander liegende Strecken mit ungefähr gleicher Länge zu erkennen. Strecken, die hintereinander liegen und deren Längen sich nur um ein relativ kleines Delta unterscheiden, bilden im Histogramm ein so genanntes Plateau (siehe Abbildung 3.5). Ein Plateau, dessen zugehörige Strecken alle kleiner sind als die doppelte mittlere Ungenauigkeit der GPS-Ortung (welche aus deren Varianz hervorgeht), ist ein Indiz dafür, dass die zu vergleichenden Tracks das gleiche Straßenstück erfasst haben, jedoch durch einen konstanten GPS-Messfehler versetzt sind.

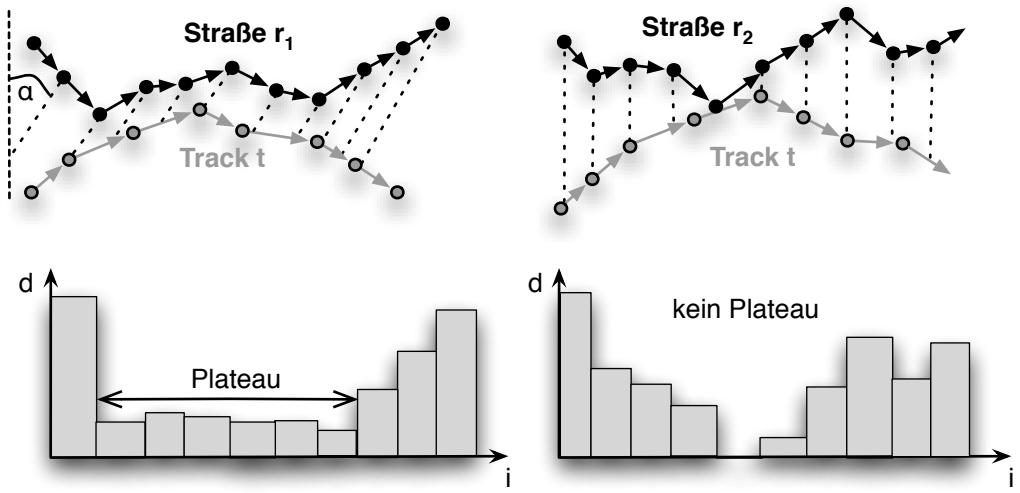


Abbildung 3.5: Erstellung von Histogrammen

Da ein Histogramm eine Vielzahl von Plateaus besitzen kann, muss das homogenste unter ihnen gefunden werden. Um jedoch mehrere Plateaus miteinander vergleichen zu können, muss zuerst der Wert eines Plateaus formalisiert werden. Generell ist ein Plateau umso besser desto länger die Anzahl seiner enthaltenen Strecken sind bzw. desto geringer die Abweichungen der Längen dieser Strecken untereinander sind. Roth formalisiert den Wert eines Plateaus mit Hilfe von Gleichung 3.1, welche wir auch im Rahmen dieser Arbeit verwenden werden.

$$v = \frac{i_1 - i_0}{d_{avg} + d_{dev}} \tag{3.1}$$

In der Gleichung stehen i_0 und i_1 für die beiden Indizes der Strecken, welche die beiden äußersten Strecken im Histogramms bilden, d_{avg} steht für den Durchschnittswert, der sich aus allen Histogrammeinträgen, die sich innerhalb der beiden Indizes befinden, ergibt und d_{dev} analog für die Standardabweichung.

Um nun das minimale Plateau zu finden, werden die Strecken zwischen allen möglichen Indizes i_0 und i_1, mit $i_0 < i_1$, im Histogramms untersucht und für die jeweiligen Strecken v berechnet (siehe Abbildung 3.6). Die beiden Werte i_0 und i_1, für die das berechnete v minimal ist, bestimmen das gesuchte Plateau. Dieses besteht aus allen Einträgen im Histogramm mit Index i, für die gilt $i_0 \leq i \leq i_1$.

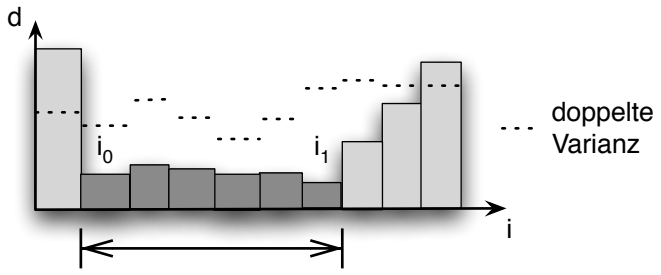


Abbildung 3.6: Berechnung des Wertes eines Plateaus

Da jede Kombination von Indizes überprüft wird, für die gilt das $i_0 < i_1$, kann die Berechnung je nach Anzahl der Wegpunkte aufwendig sein. Eine Optimierung besteht darin, ein Plateau für zwei Indizes i_0 und i_1 nur dann zu berechnen, wenn die darin befindlichen Abstandswerte kleiner sind als das Doppelte der Varianz der jeweiligen Messung. Sind ein oder mehrere Abstandswerte eines zu untersuchenden Plateaus größer als dieser Wert, so bilden die beiden GPS-Punkte nicht die gleiche Straßen ab und das Plateau kommt für eine weitere Betrachtung nicht in Frage (siehe Abbildung 3.6).

Die soeben vorgestellte Methode liefert für einen festen Winkel α ein Plateau, das durch die beiden Indizes i_0 und i_1 bestimmt ist. Da die beiden zu betrachtenden Tracks jedoch in verschiedenen Winkeln zueinander versetzt sein können, reicht eine Analyse der Histogramme für einen einzigen Winkel nicht aus. Daher müssen für zwei Tracks alle möglichen Winkel betrachtet werden, indem für jeden ein Histogramm erstellt wird. Aus der Menge dieser Histogramme wird nun das minimale Plateau über alle Histogramme bestimmt und für die folgende Fusion der Tracks verwendet. Für die Umsetzung dieses Vergleichs genügt es, zu Gunsten der Laufzeit, eine Analyse der Geraden für alle Winkel, die jeweils im Abstand von wenigen Grad zueinander liegen, durchzuführen.

Die Komplexität des vorgestellten Verfahrens liegt bei $O(t^2m^2)$, wobei t die Anzahl der erfassten GPS-Tracks und m die durchschnittliche Anzahl der Wegpunkte pro Track ist,

bei konstanter Anzahl der zu untersuchenden Winkel.

Fusion von Tracks

Durch das im letzten Abschnitt vorgestellte Vergleichsverfahren von Roth, erhalten wir zwei Indizes, mit deren Hilfe zwei Tracks fusioniert werden können. Den Ablauf dieser Fusion wollen wir nachfolgend anhand eines Beispiels vorstellen, bei dem ein Track mit einer vorhanden Straße, die von der Struktur her gesehen auch ein Track ist, fusioniert wird.

Sei t ein neu zu erfassender GPS-Track und sei WP_t das Tupel $(wp_0, \cdots, wp_n)$ aller Wegpunkte von t. Mit dem zuvor beschriebenen Verfahren wird nun für t das, für alle im Straßenmodell vorhanden Straßen, minimale Plateau in den zugehörigen Histogrammen, für verschiedene Winkel α, gesucht. Dieser Vergleich liefert ein Straße s und zwei Indizes i_0 und i_1, welches einem Plateau zwischen den Punkten wp_{i0} und wp_{i1} entspricht. Mit Hilfe dieser Indizes werden nun zwei neue Tracks t_1, mit $WP_{t1} = (wp_0, \cdots, wp_{i0})$ und t_2, mit $WP_{t2} = (wp_{i1}, \cdots, wp_n)$, gebildet (siehe Abbildung 3.7).

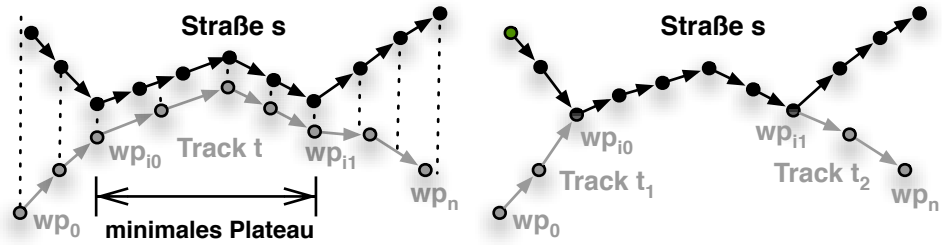


Abbildung 3.7: Teilung eines GPS-Tracks beim Vergleich mit einer Straße

Aus den Wegpunkte in $(wp_{i0}, \cdots, wp_{i1})$ und dem zugehörigen Teilabschnitt aus s werden nun, durch Interpolation der alten Koordinaten, neue Punkte errechnet, welche die entsprechenden Punkte in s ersetzen. Da die Verbindungsgerade aus t aber nicht zwangsläufig auf einem Punkt in s endet, sondern auch auf einer Kante enden kann, muss aus diesem Schnittpunkt zuerst ein neuer Punkt generiert werden, bevor mit dieser wiederum die Interpolation für den neuen Punkt durchgeführt werden kann (siehe Abbildung 3.8a). Da sich dieser Punkt auf einer Kante von s befindet, werden dessen Koordinaten und Varianz durch anteilige lineare Interpolation der beiden Eckpunkte der Kante berechnet, auf denen der Schnittpunkt liegt. Trifft die von t ausgehende Gerade direkt auf einen Knoten in t, so muss kein Schnittpunkt berechnet werden (siehe Abbildung 3.8b). Nun kann die eigentliche Interpolation zwischen diesen Schnittpunkten und den Punkten des GPS-Tracks stattfin-

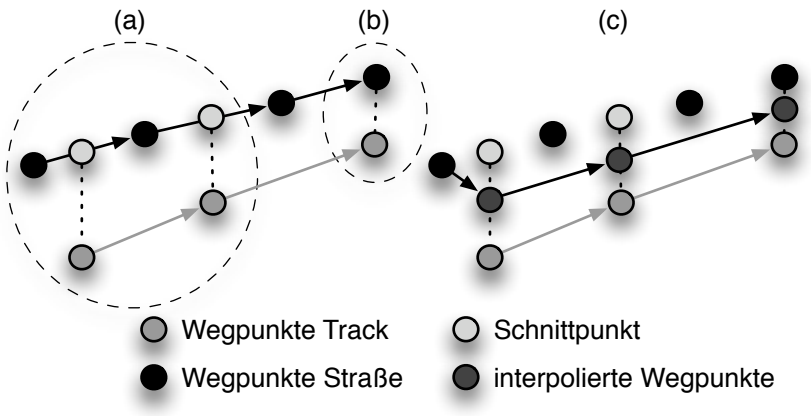


Abbildung 3.8: Interpolation der Punkte zweier GPS-Tracks

den, welche für zwei beispielhafte Punkte $P1$ und $P2$ nach folgenden Formeln erfolgt (siehe [41]):

$$
\begin{aligned}
x_{neu} &= \frac{x_{P1} \cdot var_{P2} + x_{P2} \cdot var_{P1}}{var_{P1} + var_{P2}} \\
y_{neu} &= \frac{y_{P1} \cdot var_{P2} + y_{P2} \cdot var_{P1}}{var_{P1} + var_{P2}} \\
var_{neu} &= \frac{var_{P1} \cdot var_{P2}}{var_{P1} + var_{P2}}
\end{aligned}
$$

Nach der Interpolation werden die Wegpunkte aus s, welche sich zwischen der ersten und letzten Schnittgerade aus t befinden, durch diese neuen Punkte ersetzt (siehe Abbildung 3.8c). Für die Teile des Tracks, welche nicht mit einer Straße fusioniert wurden (im Beispiel die Tracks t_1 und t_2), wird der komplette Vergleichsalgorithmus erneut aufgerufen, da diese theoretisch mit einer anderen Straße fusioniert werden können. Wenn das Vergleichsverfahren für einen Track keine Straße findet, die mit ihm ein Plateau gemeinsam hat, das klein genug ist, wird der Track als neue Straße dem Straßenmodell hinzugefügt, da er auf einer Straße aufgezeichnet wurde, die sich noch nicht im Straßenmodell befindet.

Möglichkeiten zur Optimierung

Um die Rechenlast beim Server zu reduzieren, kann die Verarbeitung eines Tracks dahingehend beschleunigt werden, dass für einen eingehenden Track nicht alle Straßen zum Vergleich herangezogen werden, sondern nur diejenigen, welche sich in der lokalen Um-

gebung des Tracks befinden. Um diese zu bestimmen, nutzen wir das in Abschnitt 3.2.1 vorgestellte Gitternetz wie folgt:

Mit jeder Straße im Straßenmodell wird eine Menge assoziiert, die alle Zellen enthält, welche von der Straße geschnitten werden. Diese Menge wird um die Menge der Nachbarzellen der einzelnen Zellen erweitert, da es sein kann, dass sich, durch die GPS-Ungenauigkeit, die Straße zu Teilen in der Nachbarzelle befindet (siehe Abbildung 3.9). Empfängt der

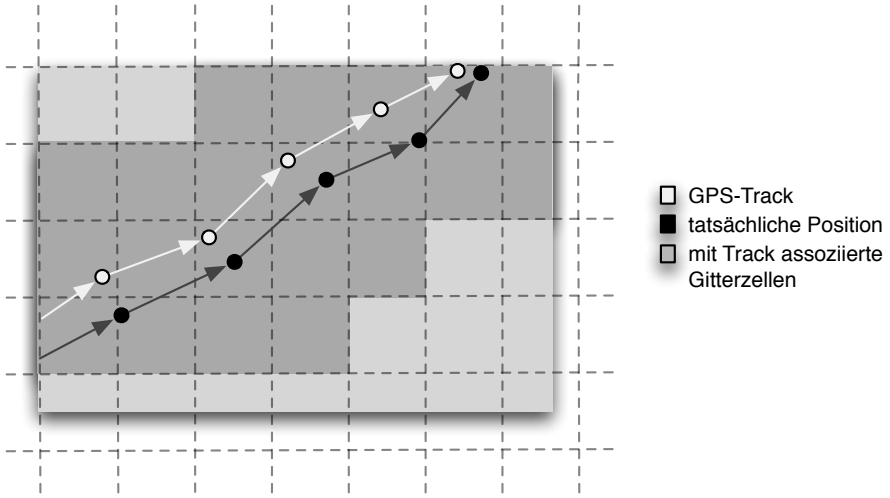


Abbildung 3.9: Verknüpfung zwischen GPS-Track und Gitterzellen

Server einen neuen Track, wird für diesen ebenfalls diese Menge bestimmt. Die Menge der Gitterzellen des Tracks wird nun nacheinander mit allen zugehörigen Mengen der gespeicherten Straßen geschnitten. Nur wenn die Menge des Tracks mit der Menge einer Straße mindestens eine Gitterzelle gemeinsam hat, wird das Vergleichsverfahren für diese beiden angewandt. Dabei genügt es den Track nur mit dem Abschnitt der Straße zu vergleichen, der in der gemeinsamen Gitterzelle liegt, anstatt die kompletten Straße zu untersuchen.

Durch dieses Verfahren, wird der Vergleich für Tracks mit Straßen unterbunden, die auf Grund ihres Abstands auf keinen Fall für eine Fusion mit dem Track in Frage kommen. Dadurch kann Rechenkapazität auf dem Server eingespart werden, falls diese kritisch ist.

Diskussion zur Straßenbreite

Zusätzlich zu der bisher vorgestellten Methode zur Verarbeitung eines GPS-Tracks, kann untersucht werden, ob es möglich ist, die Breite einer Straße aus mehreren GPS-Tracks abzuleiten. Diese Überlegung fand auch im Rahmen dieser Arbeit statt. Die folgende Analyse der Lösbarkeit des Problems kommt jedoch zu dem Ergebnis, dass eine wirklich eindeutige

Identifizierung der Straßenbreite, bei der Annahme eines durchschnittlichen Fehlers bei der GPS-Ortung, nur schwer möglich ist.

Um eine einigermaßen wahrscheinliche Aussage über die minimale Breite einer Straße machen zu können, müssen zwei GPS-Messungen vorliegen, die mit Sicherheit auf der gleichen Straße erfasst wurden und deren Kreise der Unschärfe (siehe Abschnitt 3.1.2) sich nicht überschneiden (siehe Abbildung 3.10a). In diesem Fall ist die Wahrscheinlichkeit sehr groß, dass die beiden tatsächlichen Positionen, an denen die GPS-Ortung durchgeführt wurde, nicht übereinstimmen. Dies bedeutet, dass der minimale Abstand zwischen den nicht überlappenden Kreisen als Indiz für die minimale Breite der erfassten Straße gesehen werden kann. Überlappen sich die Kreise hingegen, so können die beiden GPS-Ortungen mit hoher Wahrscheinlichkeit auch am gleichen Ort durchgeführt worden sein, woraus sich keine Aussage über die Breite der Straße schließen lässt (siehe Abbildung 3.10b).

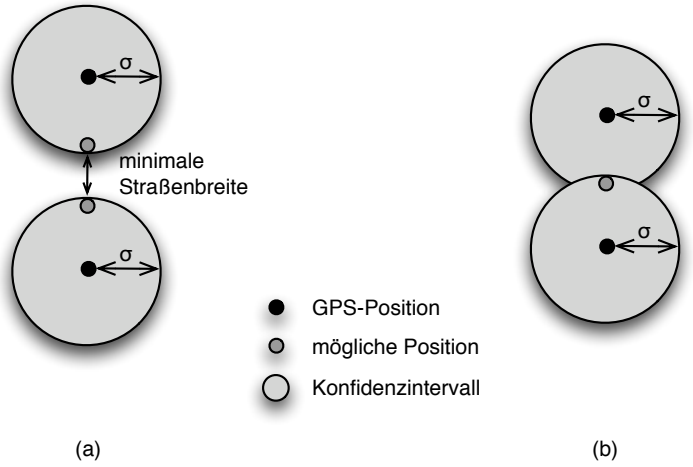


Abbildung 3.10: Mögliche Positionen bei der Bestimmung der Straßenbreite

Das Problem dabei ist nun, zu entscheiden, wann zwei GPS-Positionen potentiell auf der gleichen Straße erfasst wurden, und somit eine Aussage über die Straßenbreite gemacht werden kann, und wann die GPS-Ortungen zwei verschiedene, parallele Straßen beschreiben. Gehen wir davon aus, dass eine Straße eine maximale Breite von 10 Metern hat. Ist eine Straße nun jedoch, wie in Abbildung 3.11, deutlich kleiner und existiert eine nicht weit entfernte Parallelstraße, so ist es ohne Kenntnisse der genauen Straßenverläufe nicht möglich, zu entscheiden, ob aus den beiden Punkten eine Aussage über die Straßenbreite gemacht werden kann oder nicht. Gehen wir von einem Stadtgebiet aus, indem der GPS-Fehler durch Reflektionen der Gebäude im Schnitt bis zu 20 Meter hoch sein kann, so müsste ein möglicher Algorithmus zum Vergleich von GPS-Tracks noch GPS-Ortungen berücksichtigen die 50 Meter auseinander liegen und somit, speziell in einem Stadtgebiet,

relativ häufig tatsächlich auf verschiedenen Straßen erfasst wurden.

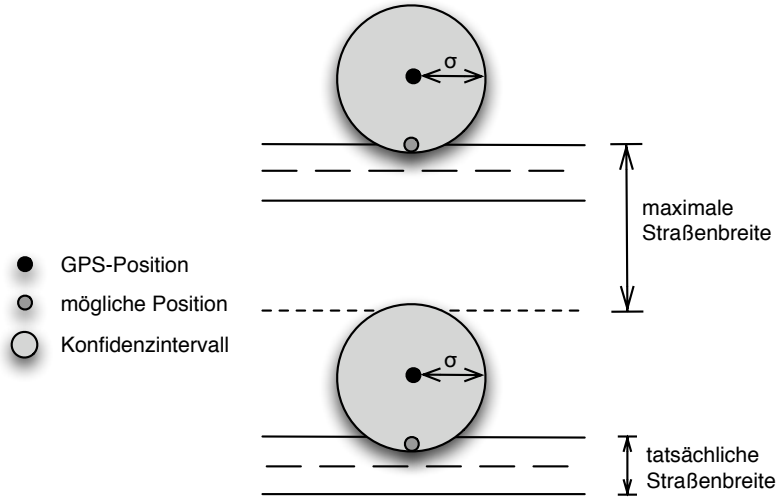


Abbildung 3.11: Problemfall bei der Bestimmung der Straßenbreite

Da diese Betrachtung aufgrund der gefundenen Probleme offensichtlich eine differenziertere Analyse benötigt, welche nicht in den Rahmen dieser Arbeit passen würde, wird in der späteren Umsetzung auf die Ermittlung der Straßenbreite verzichtet.

3.3 Qualitätsmodell

Wie bereits in der Zielsetzung erwähnt wurde, besteht ein wesentlicher Teil dieser Arbeit darin, ein Qualitätsmodell zu entwickeln, mit dessen Hilfe beurteilt werden kann, wie gut eine Straße oder ein Teilbereich des Zielgebiets bisher erfasst wurde. Werden diese Informationen in quantitativer Form bereitgestellt, können sie direkt vom Server zur gezielten Steuerung der Erfassung genutzt werden. Auf Basis dieser Informationen ist es dem Server möglich, mit Hilfe von dynamischen Erfassungsalgorithmen (siehe Abschnitt 3.4), auf Veränderungen bei diesen Qualitätswerten zu reagieren und den Fokus der Erfassung so zu ändern, dass eine Ressourceneinsparung auf Seiten der Mobilgeräte möglich ist. Die Schwierigkeit bei der Definition eines solchen Modells beruht darin, dass keine Referenzdaten vorliegen, welche die Verläufe der tatsächlichen Straßen enthalten, da mit deren Hilfe eine relativ einfache Aussage über die Qualität der erfassten Straßen gemacht werden könnte (siehe zum Beispiel [45]). Es muss also ein Qualitätsmodell definiert werden, dass nur anhand der bisher erfassten Daten eine Qualitätsaussage über diese zulässt.

Dieses Qualitätsmodell soll in diesem Abschnitt Stück für Stück entwickelt werden. Da-

zu beginnen wir in Abschnitt 3.3.1 mit der Definition einer Qualitätsmetrik, welche eine Qualitätsaussage über eine einzelne Straße ermöglicht. Darauf aufbauend wollen wir einen Qualitätswert für einzelne Gitterzellen, des in Abschnitt 3.2.1 eingeführten Gitternetzes, bestimmen. Um jedoch auf Basis der Metrik für einzelne Straßen Aussagen über den Qualitätszustand einer einzelnen Gitterzelle des Zielbereichs machen zu können, müsste dem System a priori bekannt sein, wie viele Straßen sich in dieser Gitterzelle insgesamt befinden, um berücksichtigen zu können, wie viele Straßen noch nicht erfassten wurden. Da das System jedoch zu keinem Zeitpunkt Kenntnis über Struktur und Anzahl der tatsächlich vorhanden Straßen hat, ist eine solche Aussage unmöglich. Daher muss eine Annäherung getroffen werden, welche eine ungefähre Aussage darüber geben kann, wie viel Prozent der in der Gitterzelle vorhandenen Straßen bereits vom System erfasst wurden. Diese Metrik wird in Abschnitt 3.3.2, als die Definition des Zeitwerts der Erfassung, eingeführt. Mit Hilfe der Qualitätsmetrik für einzelne Straßen und dem Zeitwert der Erfassung kann nun in Abschnitt 3.3.3 eine Qualitätsaussage für eine einzelne Gitterzelle definiert werden, anhand welcher der Server den anschließend definierten Erfassungsalgorithmus steuern kann.

3.3.1 Qualität einzelner Straßen

Um die Qualität der im Straßenmodell erfassten Daten beurteilen zu können, ist es notwendig, eine Qualitätsmetrik für einzelne Straßen zu definieren. Diese soll für jede Straße im Straßenmodell, unter Berücksichtigung der bisher erfassten Wegpunkte, einen Vergleich der Straßen untereinander ermöglichen. Dabei gilt es zu berücksichtigen, dass es verschiedene Kriterien gibt, nach denen die Qualität einer Straße beurteilt werden kann. So kann zum einen eine Aussage darüber gemacht werden, wie gut die Form des Straßenverlaufs durch die vorhandenen Wegpunkte beschrieben wird und zum anderen kann versucht werden zu beurteilen, wie gut die Genauigkeit der Koordinaten der Wegpunkte im Modell ist. Für beide Metriken werden wir nachfolgend eine geeignete Definition treffen. Anschließend wird kurz diskutiert, wie aus diesen beiden unabhängigen Metriken eine einzige Metrik gebildet werden kann, welche für jede Straße r einen eindeutigen Qualitätswert liefert.

Genauigkeit des Straßenverlaufs

Das in Abschnitt 3.2.1 vorgestellte Modell des Straßenverlaufs hat die inhärente Eigenschaft, dass Kurven nur annäherungsweise abgebildet werden können, da diese durch Punkte und Kanten modelliert werden. Angenommen ein relativ schnell fahrendes Fahrzeug erfasst GPS-Positionen während es eine Kurve durchfährt. Da die erfassten Wegpunkte aufgrund der Geschwindigkeit relativ weit auseinander liegen, ist eine genaue Beschreibung des Straßenverlaufs nicht mehr gegeben. Weil die Mobilgerät sich außerdem mit unter-

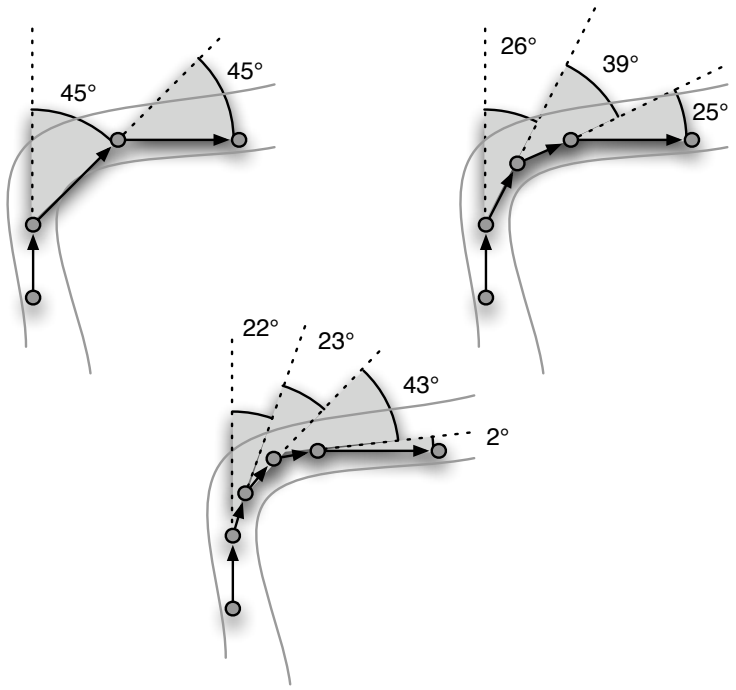


Abbildung 3.12: Kantenwinkel bei der Genauigkeit des Kurvenverlaufs

schiedlichen Geschwindigkeiten fortbewegen, können so größere Unterschiede bezüglich der Genauigkeit der Beschreibung des Straßenverlaufs entstehen.

Unter der Annahme, dass der GPS-Messfehler für einen erfassten Track relativ konstant ist, gilt der Grundsatz, dass je mehr Wegpunkte für eine Kurve erfasst wurden, desto genauer ist die Beschreibung des tatsächlichen Kurvenverlaufs. Im Vergleich dazu kann eine beliebig lange Gerade durch zwei Punkte exakt beschrieben werden. Nehmen wir an, dass die aufgezeichneten Tracks aus Geraden und Kurven bestehen. Zwar kann man argumentieren, dass Straßenverläufe zusätzlich Abzweigungen und Kreuzungen aufweisen, dabei ist jedoch zu beachten, dass diese von Fußgängern und Autos ebenfalls in einer Kurvenform passiert werden und somit aus einem GPS-Track nicht ersichtlich sind.

Um nun die Beschreibungsgenauigkeit einer Kurve zu ermitteln, schauen wir uns die Größe der Winkel an, die zwischen den Kanten, welche die Kurve beschreiben, liegen. Dabei gilt, dass eine ungenaue Darstellung, durch wenige Punkte, in großen Winkeln zwischen den Kanten dieser Punkte resultiert. Im Gegenzug ist die Darstellung umso genauer, je kleiner die Winkel zwischen diesen Kanten sind (siehe Abbildung 3.12). Mithilfe dieser Winkel, können wir nun eine Aussage über die Genauigkeit der Form einer Straße im Straßenmodell treffen.

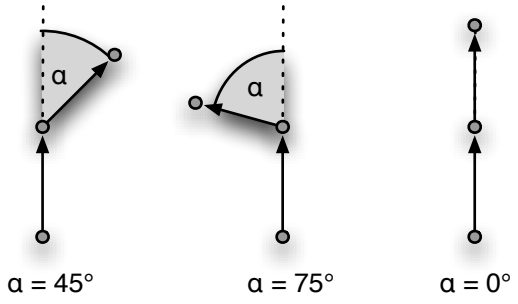


Abbildung 3.13: Definition des Winkel α

Gegeben sei eine bereits erfasste Straße r mit den Wegpunkten P_0 bis P_N und den daraus resultierenden Kanten E_0 bis E_{N-1}. Wir definieren α_i als den Betrag des kleineren Winkels, den die Verlängerung von E_i und E_{i+1} aufspannt (siehe Abbildung 3.13). Zuerst definieren wir für r den Mittelwert über alle Winkel α:

$$A_v(r) = \frac{1}{N-2} \cdot \sum_{i=0}^{N-2} \alpha_i \tag{3.2}$$

Im Idealfall sind alle Kurven der Straße durch so viele Punkte beschrieben, dass die Winkel zwischen den Kanten der Straßen nur wenige Grad aufspannen. Eine Gerade wird durch senkrecht aufeinander stehende Kanten abgebildet, welche einen Winkel von null Grad aufspannen, und daher durch dieses Modell als absolut genau klassifiziert wird. Ein relativ geringer Wert für A_v beschreibt daher eine gute Genauigkeit des Straßenverlaufs. Das bedeutet aber auch, dass A_v mit steigender Winkelsumme wächst. Da dies aber kontraintuitiv für ein Qualitätsmaß ist, definieren wir die Qualität der Genauigkeit eines Straßenverlaufs als:

$$Q_v(r) = \frac{1}{A_v(r)} \tag{3.3}$$

Da es für den Wert von Q_v theoretisch keine Obergrenze gibt, ist mit diesem Maß lediglich einen Vergleich von einer Menge von Straßen untereinander möglich. Eine absolute Metrik, um Aussagen über einzelne Straßen unabhängig von anderen treffen zu können, ist aufgrund der Unkenntnis des genauen Straßenverlaufs und des GPS-Messfehlers nur schwer möglich. Da der Server bei der Entscheidung, welche Teile des Zielbereichs zu erfassen sind, jedoch eine relative Entscheidung treffen muss, genügt es, wenn die Metrik eine relative Aussage liefert.

Genauigkeit der Straßenposition

Nachdem wir bei der Genauigkeit des Straßenverlaufs eine Straße als Gesamtkonstrukt untersucht haben, wollen wir nun beurteilen, wie genau die Positionen der einzelnen Wegpunkte einer Straße, hinsichtlich ihrer absoluten Position, erfasst wurden. Dazu schauen wir uns an, wie oft ein Wegpunkt im Straßenmodell durch die Interpolation mit verschiedenen Tracks angepasst wurde und in wie weit diese Positionen übereinstimmen. Hierbei spielt die für jeden Wegpunkt erfasste Varianz eine entscheidende Rolle, da diese eine Wahrscheinlichkeit angibt, wie weit der gemessene Wert von der tatsächlichen Position abweicht.

Wird bei dem in Abschnitt 3.2.2 genutzten Verfahren von Roth ein neuer Punkt P_{neu} durch Interpolation zweier alten Punkte P_1 und P_2 generiert, so erbt dieser auch die Varianzen der beiden alten Punkte, durch folgende, auf Bayes bedingter Wahrscheinlichkeiten beruhender, Gleichung:

$$var_{P_{neu}} = \frac{var_{P_1} \cdot var_{P_2}}{var_{P_1} + var_{P_2}} \tag{3.4}$$

Durch Vergleich von $var_{P_{neu}}$ mit den alten Varianzen können wir leicht sehen, dass die neu entstandene Varianz kleiner ist, als die beiden alten. Wir zeigen für die Varianz des Punkts P_1 (der Fall P_2 ist analog):

$$\begin{aligned}
var_{P_1} &> var_{P_{neu}} \\
var_{P_1} &> \frac{var_{P_1} \cdot var_{P_2}}{var_{P_1} + var_{P_2}} \\
var_{P_1} \cdot (var_{P_1} + var_{P_2}) &> var_{P_1} \cdot var_{P_2} \\
var_{P_1}^2 + var_{P_1} \cdot var_{P_2} &> var_{P_1} \cdot var_{P_2} \\
var_{P_1}^2 &> 0
\end{aligned}$$

Was offensichtlich für alle $\mathbb{R} \setminus \{0\}$ gilt.

Somit wird die Varianz eines Punktes durch jede Interpolation kleiner und die Wahrscheinlichkeit steigt, dass sich die interpolierten Koordinaten an dem tatsächlichen Straßenpunkt befinden. Diese Varianz dient uns daher als Qualitätsmaß für die Genauigkeit der Position eines Punktes: Je geringer die Varianz eines Punktes, desto besser ist dessen Genauigkeit.

Wir definieren zuerst den Durchschnitt der Varianzen für eine Straße r als den Durchschnitt

der Varianzen der einzelnen Punkte P_i:

$$A_p(r) = \frac{1}{|P|} \cdot \sum_{i=1}^{|P|} var_{P_i} \tag{3.5}$$

Da die Genauigkeit der Punkte umso besser ist, je kleiner der Wert von A_p ist, definieren wir die Qualität der Genauigkeit einer Straße r als deren Kehrwert:

$$Q_p(r) = \frac{1}{A_p(r)} \tag{3.6}$$

Die Varianz eines Wegpunkts wird mit jeder Interpolation kleiner und läuft daher gegen null. Q_p kann dadurch theoretisch beliebig groß werden.

Diese Qualitätsmetrik lässt somit den Qualitätswert einer Straße steigen, umso mehr Punkte auf dieser erfasst wurden. Das lässt sich so erklären, dass umso mehr Tracks auf der gleichen Straße erfasst wurden, desto geringer ist die Varianz der Wegpunkte dieser Straße, da jedes Mal Interpolationen zwischen diesen Tracks stattfinden, die den Qualitätswert Q_p erhöhen. Dadurch steigt die Qualität der erfassten Straße.

Integration beider Metriken

Für den späteren Verlauf der Arbeit, ist es notwendig, für jede Straße ein eindeutiges Qualitätsmaß zu verwenden, anhand dessen das System Entscheidungen über den Fortgang der Erfassung treffen kann. Da wir bisher jedoch für die Qualität einer Straße zwei Metriken definiert haben, ist eine Integration dieser beider Metriken notwendig. Ziel ist es im Wesentlichen eine Funktion zu finden, die als Eingabe die oben definierten Qualitätsmetriken nimmt und daraus einen einzigen, aussagekräftigen Wert bildet. Nachfolgend sollen zwei solcher Funktionen vorgestellt und ihre Charakteristiken untersucht werden.

Als erstes betrachten wir eine einfache Funktion, die als Funktionswert das Produkt der beiden Eingangsmetriken ausgibt:

$$Q_{TP}(r) = Q_v(r) \cdot Q_p(r) \tag{3.7}$$

Wenn wir uns den Graphen dieser Funktion anschauen (siehe Abbildung 3.14), so ist ersichtlich, dass Q_{TP} nur dann einen hohen Wert annimmt, wenn beide Eingangswerte größer als null sind. Sind beide Eingangswerte relativ groß, so steigt der Funktionswert überproportional an.

Bei dieser Funktion müssen wir uns zusätzlich überlegen, welche Werte die beiden Ein-

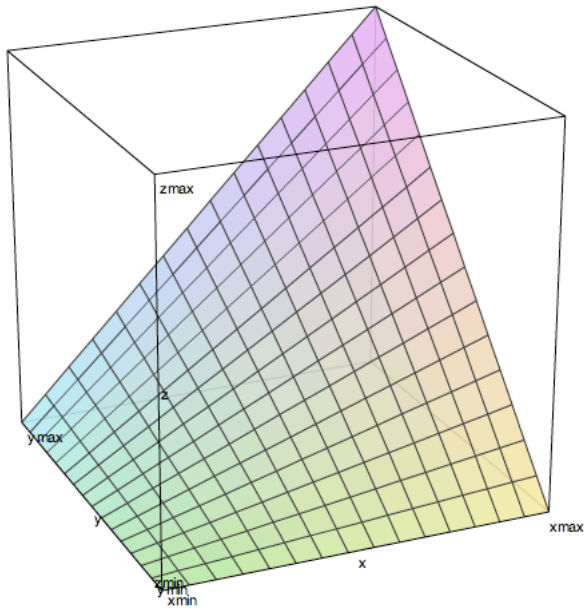


Abbildung 3.14: Resultierende Straßenqualität bei Verwendung der Produkt-Funktion

gangsfunktionen haben können. Während $Q_p(r)$ mit ziemlicher Sicherheit kleiner wie 1 ist, da die Varianz der Messungen so gut wie nie im Schnitt kleiner als eins sein wird, kann $Q_v(r)$ deutlicher kleiner als eins werden. Dies ist zum Beispiel der Fall, wenn die Zwischenwinkel einer geraden Straße berechnet werden. Da sich daraus ein überproportional großer Wert für $Q_{TP}(r)$ ergeben würde, der den Wert von $Q_p(r)$ bei weitem übersteigt, beschränken wir die Funktion $Q_v(r)$ wie folgt:

$$Q_{TP}(r) = \min(Q_v(r), 1) \cdot Q_p(r) \tag{3.8}$$

Die zweite Funktion die wir uns anschauen, gibt den Minimalwert beider Eingangsparameter zurück und ist wie folgt definiert:

$$Q_{TM}(r) = \min(c \cdot Q_v(r), Q_p(r)) \tag{3.9}$$

Da die beiden Qualitätsmetriken $Q_v(r)$ und $Q_p(r)$ nicht direkt miteinander verglichen werden können, kann eine Konstante c zur Dimensionierung herangezogen werden. Mithilfe dieser Konstante kann der Systembenutzer eine stärkere Gewichtung für eine der beiden Metriken in das System einfließen lassen. Wählt er zum Beispiel $c = 0.5$, so wird bei konstantem $Q_p(r)$ der gleiche Ausgangswert nur erreicht, wenn $Q_v(r)$ doppelt so groß ist, wie im Fall $c = 1$.

Der Unterschied der beiden Metriken, liegt in ihrem Verhalten bei großen Eingangswerten. Während die erste Gleichung für große Werte in ein überproportionales Wachstum übergeht, bleibt die zweite Funktion linear. Die erste Funktion begünstigt also Szenarios, bei denen beide Eingangs-Qualitätsmetriken gut sind, besonders stark, während die zweite Funktion ein solches Szenario nicht überdurchschnittlich stark betont. Die erste Funktion ist daher eher geeignet, wenn dem Systembenutzer wichtig ist, das beide Metriken der Straßenqualität einen hohen Wert haben. Für die später folgende Implementierung wählen wir die Produkt-Funktion.

3.3.2 Zeitwert der Erfassung

Neben der Qualitätsmetrik für eine einzelne Straße, ist für den späteren Verlauf der Arbeit eine zweite Qualitätsmetrik von wichtiger Bedeutung — der Zeitwert der Erfassung. Nachfolgend werden wir die Einführung dieses Werts motivieren und erklären, warum dieser von so wichtiger Bedeutung ist, bevor dann die Definition dieses Werts erfolgt.

Motivation

Das Ziel dieser Masterarbeit liegt darin, die Straßenerfassung möglichst effizient zu gestalten, um so die Ressourcen der Mobilgeräte zu schonen und dadurch die Akzeptanz der Benutzer dieses opportunistischen Verfahrens zu erhöhen. Um dies zu erreichen, könnte man sich überlegen, anhand einzelner Straßen zu entscheiden, ob ein Gerät die Erfassung auf dieser Straße fortsetzen soll oder ob diese bereits so gut erfasst wurde, dass das Gerät die Erfassung für diese Straße unterdrücken kann, um somit Ressourcen einzusparen. Dazu müsste das Gerät dem Server eine eindeutige Informationen darüber geben, welche Straße es momentan erfasst. Der Server würde nun anhand seines Straßenmodells entscheiden, ob diese Straße bereits ausreichend gut erfasst wurde und dies dem Mobilgerät mitteilen. Dieser einfach klingende Ansatz bringt jedoch eine Reihe von Problemen mit sich. Will ein Mobilgerät lokal entscheiden, ob es sich auf einer bereits erfassten Straße befindet oder nicht, müsste es alle bereits erfassten Straßen des Straßenmodells kennen. Soll diese Entscheidung anstatt dessen vom Server getroffen werden, müsste das Mobilgeräte die erfassten Wegpunkte in Echtzeit an den Server schicken. Beide Ansätze skalieren offensichtlich nicht. Hinzu kommt das Problem, dass weder Mobilgerät noch Server aufgrund der Unschärfe der GPS-Ortungen feststellen können, auf welcher Straße sich das Mobilgerät gerade befindet und ob diese schon erfasst wurde. Wie in Abschnitt 3.2 deutlich wurde, können nur komplett vorliegende Tracks daraufhin untersucht werden, ob sie auf der gleichen Straße aufgezeichnet wurden. Da dieser Track aber noch nicht vorliegt, wenn ein Mobilgerät gerade erst damit begonnen hat eine Straße zu erfassen, muss ein

anderes Verfahren gefunden werden, um beurteilen zu können, ob ein Mobilgerät einen Bereich erfassen soll oder die Erfassung zu Gunsten der Ressourcenschonung einstellen soll. Wir wollen daher das in Abschnitt 3.2.1 definierte Gitternetz nutzen, um Aussagen über einzelne Gitterzellen zu treffen und somit Einfluss auf die Erfassung nehmen zu können.

Da die bisherige Qualitätsmetrik nur einen Wert für eine einzelne Straße liefert, ist eine Definition der Qualität einer Gitterzelle anhand dieser Metrik nicht möglich. Nehmen wir zum Beispiel an, dass in einer Gitterzelle bisher nur eine einzige Straße von einem Mobilgerät sehr gut erfasst wurde, aber noch viele andere Straßen in dieser Gitterzelle vorhanden sind, welche noch nicht entdeckt wurden. Es würde daher keinen Sinn ergeben, der Gitterzelle einen hohen Qualitätswert zuzuweisen, da die Mehrheit der Straßen noch gar nicht entdeckt wurde und die Qualität der Gitterzelle daher eher als schlecht beurteilt werden sollte. Wir benötigen also ein Maß, welches uns angibt, welcher Anteil der Straßen bisher entdeckt wurden. Da nun aber eine genaue Beurteilung dieses Anteils nicht möglich ist, da die genaue Anzahl der im Weltmodell befindlichen Straßen nicht bekannt ist, kann über ihn nur eine Abschätzung getroffen werden. Um diese Abschätzung zu ermöglichen definieren wir nachfolgend ein Maß für den Zeitwert der Erfassung, welcher zu jedem Zeitpunkt eine Aussage darüber machen soll, wie weit die Erfassung bisher fortgeschritten ist. Unter Berücksichtigung dieses Wertes, kann eine Definition für den Qualitätswert einer einzelnen Zelle getroffen werden, welche den Qualitätswert der einzelnen Straßen einer Gitterzelle in Bezug zu diesem Zeitwert relativiert. Diese Metrik kann, wie bereits erwähnt, nur eine Abschätzung sein und stellt eine Annäherung an den tatsächlichen Grad der Vollständigkeit der Straßenerfassung dar, welcher den prozentualen Anteil der im Weltmodell vorhandenen Straßen zu den im Straßenmodell endeckten angibt.

Eigenschaften

Der Zeitwert stellt eine Annäherung daran da, wie viel Prozent der im gesamten Modell vorhandenen Straßen durch die Mobilgeräte bisher erfasst wurden. Für die Erstellung eines Grundmodells für den Zeitwert, sollen daher folgende Annahme dienen:

(i) Zu Beginn der Kartengenerierung liegt der Zeitwert bei null.

(ii) Der Zeitwert wächst mit der Zeit t bis zu einer Schranke S, welche mit der Entdeckung aller Straßen gleichzusetzen ist.

(iii) Die Schranke S wird weder überschritten noch jemals vollständig erreicht.

Während Annahme (i) offensichtlich ist, da zu Beginn noch keine Straßen erfasst wurden, stecken hinter den anderen Bedingungen folgende Überlegungen: Sobald die Straßenerfassung durch die mobilen Sensoren begonnen hat, werden neu gefundene Straßen sofort in das

Straßenmodell übertragen, da dieses noch leer ist und jede Straße bisher noch unbekannt ist. Mit Fortschreiten der Zeit, wächst die Zahl der im Modell erfassten Straßen immer weiter und es tritt schließlich eine Sättigung ein, da die Mehrheit der Straßen entdeckt und bereits im Modell vorhanden ist. Der Zuwachs an neu entdeckten Straßen wird geringer und nähert sich asymptotisch der Schranke S. Dieses Verhalten spiegelt Bedingung (ii) wider. Auch wenn die Erfassung schon eine geraume Zeit läuft und der Zeitwert bereits ziemlich nahe an S liegt, kann nie angenommen werden, dass schon wirklich alle Straßen erfasst wurden, da es immer Straßen geben kann die von keinem Mobilgerät passiert wurden. Dieser Gedanken wird durch Bedingung (iii) ausgedrückt.

Umsetzung

Um diese Bedingungen in einer mathematischen Funktion auszudrücken, wählen wir eine einfache Form des beschränkten Wachstums, welche die geforderten Eigenschaften erfüllt. Dafür definieren wir den Zeitwert ZW zum Zeitpunkt t als:

$$ZW(t) = 1 - e^{-a \cdot t} \quad mit \quad a > 0 \quad und \quad t \geq 0 \tag{3.10}$$

Für die Schranke S wählen wir $S = 1$. Der Parameter a muss noch bestimmt werden. Dieser spielt für das Wachstum von Funktion (3.10) eine entscheidende Rolle. Wie Abbildung 3.15 zeigt, erreicht die Funktion für ein großes a schneller die Schranke S, während das Wachstum desto geringer ist, je kleiner a ist. Da a der entscheidende Parameter ist, wollen wir daher nachfolgend versuchen eine geeignete Dimensionierung zu treffen.

Parameterwahl

Die beste Belegung für Parameter a wäre dann gefunden, wenn Funktion 3.10 den Fortschritt der Erfassung genau abbilden würde. Wie bereits erwähnt ist diese Belegung jedoch auf Grund der Unkenntnisse hinsichtlich des tatsächlichen Straßennetz, sowie der Verteilung und Bewegungsmuster der Mobilgeräte, a priori nicht zu ermitteln. Daher legen wir für die Belegung des Parameters a eine Konvention fest, welche unter Berücksichtigung der gesamten Dauer der Erfassung diesen entsprechend dimensioniert.

Angenommen die Erfassung endet nach einer Zeitdauer von t_n Zeiteinheiten und beginnt zum Zeitpunkt null. Wir wählen a so, dass für den Zeitwert der Erfassung gilt:

$$ZW(t_n) = 1 - e^{-a \cdot t_n} = 0.99 \tag{3.11}$$

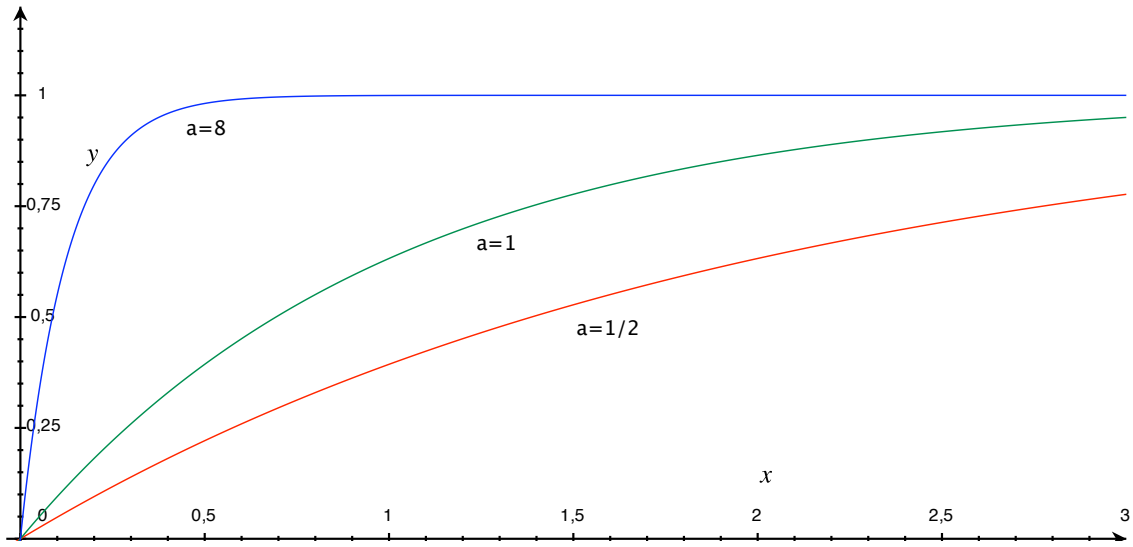


Abbildung 3.15: Zeitwert-Funktion für verschiedene Steigungsparameter

Durch Umstellen nach a erhalten wir folgende Gleichung:

$$a = -\frac{\ln(0.01)}{t_n} \tag{3.12}$$

Wir wählen also a so, dass der Zeitwert zu Ende der Erfassungszeit den Wert 0.99 erreicht (den Wert 1 können wir nicht wählen, da er nie erreicht wird). Unter der Annahme, dass der Benutzer des Systems, aufgrund seiner ortsspezifischen Vorkenntnisse des Zielbereichs, die Erfassung ausreichend lange laufen lässt, sind zu Ende der Erfassung mit ziemlich hoher Wahrscheinlichkeit alle Straßen entdeckt.

Abschließend soll zur Verdeutlichung der Definition des Qualitätswerts folgendes Beispiel dienen. Wir gehen davon aus, dass die gesamte Straßenerfassung über einen Zeitraum von zehn Stunden abläuft, t_n hat somit den Wert von 36 000 Sekunden. Durch Einsetzen in Gleichung 3.3.2 erhalten wir für a den gerundeten Wert 0.000128 und somit nachstehende Gleichung, die für $t = 36\,000$ den Wert 0.99 ausgibt (gerundet).

$$ZW(t) = 1 - e^{-0.000128 \cdot t}$$

Da dieser Zeitwert nur eine Abschätzung der Erfassung auf Basis der abgelaufenen Zeit gibt, kann überlegt werden, in diese Definition weitere Systemeigenschaften einzubauen und somit diese Abschätzung zu verbessern. Eine Möglichkeit wäre die Anzahl der zurzeit

im Zielbereich befindlichen Mobilgeräte zu integrieren, da eine höhere Anzahl von Geräten die Erfassung entsprechend beschleunigt. Ein anderer Ansatz wäre, das Delta der Qualitätszunahme der einzelnen Gitterzellen zu berücksichtigen, wobei die Annahme gilt, dass eine hohe Qualitätszunahme einen frühen Zeitpunkt der Erfassung widerspiegelt, während eine langsame Zunahme ein Indikator für eine bereits weit fortgeschrittene Erfassung ist. Da eine Umsetzung dieser Vorschläge jedoch zuerst weitere Überlegungen, bezüglich der Dynamik der Bewegungsabläufe in einem konkreten Szenario, und eine anschließende Evaluation benötigen, beschränken wir uns in dieser Arbeit auf die in Gleichung 3.10 getroffene Definition und werten die vorgestellten Ideen als Ausblick für zukünftige Betrachtungen.

3.3.3 Qualität einzelner Gitterzellen

Damit der in Abschnitt 3.4 folgende Erfassungsalgorithmus so arbeiten kann, dass sich die Erfassung durch die Mobilgeräte auf bisher qualitativ schlecht erfasste Gitterzellen konzentriert, muss dieser über Qualitätsdaten für jede einzelne Gitterzelle verfügen. Durch die Definition einer Qualitätsmetrik für Straßen und die Einführung eines Zeitwerts, kann diese Definition nun getroffen werden.

Dazu betrachten wir die Menge der in Gitterzelle $C_{x,y}$ verlaufenden Straßenabschnitte $R_{x,y}$ und bilden den Durchschnitt ihrer jeweiligen Qualität. Auch wenn die Qualität einer Straße in Abschnitt 3.3.1 für jeweils eine komplette Straße definiert wurde, lässt sich diese Definition ohne weiteres auf einen Teilabschnitt dieser Straße reduzieren, indem nur die Punkte und Kanten betrachtet werden, welche sich innerhalb der Gitterzelle $C_{x,y}$ befinden. Das daraus resultierende Qualitätsmaß bezeichnen wir mit $Q_r(x, y)$. Durch Multiplikation des Zeitwerts ZW, der stets kleiner wie eins ist, wird nun $Q_r(x, y)$ gedämpft, um die Tatsache zu berücksichtigen, dass wahrscheinlich noch nicht alle Straßen in der Gitterzelle entdeckt wurden. Daraus ergibt sich folgenden Gleichung, welche für einen Zeitpunkt t den Qualitätswert $Q(t, x, y)$ einer Gitterzelle $C_{x,y}$ folgendermaßen bestimmt:

$$Q(t, x, y) = ZW(t) \cdot Q_r(x, y) \tag{3.13}$$

Mit Hilfe dieser Formel, steuert der Server die Erfassung. Welche konkreten Algorithmen er verwendet, um den Qualitätswert einer Gitterzelle zu berücksichtigen, zeigt der folgende Abschnitt, welcher sich mit der Steuerung und der gezielten Erfassung von Gitterzellen beschäftigt.

3.4 Steuerung der Erfassung

Um festzulegen, wann welches Mobilgerät Sensordaten erfassen soll und wann diese an den zentralen Server übermittelt werden, muss ein Algorithmus gefunden werden, der die Mobilgeräte hinsichtlich dieser Entscheidungen steuert, da diese einen großen Einfluss auf Aktualität und Vollständigkeit des Gesamtmodells, sowie auf den Energieverbrauch der Mobilgeräte, haben. Dieser Algorithmus muss sich selbstständig an den Zustand des Straßenmodells anpassen und qualitativ schlecht erfasste Bereiche in den Fokus der Erfassung rücken, damit bereits gut erfasste Bereiche vernachlässigt werden können, um so die Ressourcen der Mobilgeräte zu schonen. Ziel dieses Abschnitts ist es, einen solchen Algorithmus zu definieren.

Bevor dieser Algorithmus definiert werden kann, müssen jedoch zuerst grundlegende Mechanismen definiert werden, die festlegen, wie der Server die verfügbaren Mobilgeräte verwaltet, dies geschieht in Abschnitt 3.4.1, und wie er die gegenwärtigen Standorte der Mobilgeräte intern verwaltet (Abschnitt 3.4.2). Danach beginnt die eigentliche Definition des Erfassungsalgorithmus, welche von der in Abschnitt 3.4.4 definierten Initialisierung der Erfassung bis Abschnitt 3.4.7 reicht. Danach wollen wir in Abschnitt 3.4.8 das Verhalten dieses Erfassungsalgorithmus an einem Beispiel verdeutlichen, bevor wir in Abschnitt 3.4.9 die Möglichkeit untersuchen, aus den ermittelten Erfassungsinformationen Aussagen über die Art der Straße zu schließen. Zum Abschluss stellen wir in Abschnitt 3.4.10 Optimierungspotentiale vor, durch die weitere Verbesserungen erzielt werden können.

3.4.1 Verwaltung der Mobilgeräte

Um bestimmen zu können, ob ein Mobilgerät gegenwärtig GPS-Positionen erfasst oder nicht, unterscheidet der Server für alle am System teilnehmenden Mobilgeräte die drei Zustände aktiv, passiv und ausgeschaltet. Ein Gerät wird als passiv betrachtet, wenn es angeschaltet ist, aber vorübergehend keine GPS-Daten erfasst. Aktive Geräte hingegen sammeln GPS-Daten und senden diese in regelmäßigen Abständen an den Server. Während der Server aktive und passive Geräte mit ihren aktuellen Zuständen in einer Liste verwaltet, um diese gegebenenfalls gezielt kontaktieren zu können, speichert er keinerlei Informationen über Mobilgeräte die ausgeschaltet sind. Ändert ein Gerät seinen Status, so schickt es ein Update an den Server, wie wir in den folgenden Abschnitten sehen werden.

Ein Gerät, das beim Server als passiv geführt wird, trägt nicht zur Erfassung bei. Dafür gibt es im Wesentlichen zwei Gründe:

- Ein Gerät erfasst keine Daten, weil es sich außerhalb des Zielbereichs oder in einer Zelle befindet, die vom Server aus Gründen der Ressourcenschonung vorübergehend

von der Erfassung ausgeschlossen wurde. Dazu in den folgenden Abschnitten mehr.

- Ein Gerät befindet sich innerhalb eines Gebäudes, empfängt daher kein GPS-Signal und kann deswegen keine GPS-Wegpunkte erfassen. Das Gerät kann diesen Zustand jedoch, durch das fehlende GPS-Signal, relativ leicht erkennen. Es meldet sich daraufhin beim Server als passiv.

3.4.2 Lokationsmanagement

Viele Kommunikationssystemen bestehen sowohl aus mobilen Einheiten, wie auch aus statischen Infrastrukturelementen, wie zum Beispiel einem Server. Dabei ist es oft nötig, dass der Server die Positionen der Mobilgeräte kennt, um ihnen positionsabhängige Informationen zu übertragen. Wenn die Positionsinformationen jedoch direkt durch die Mobilgeräte erfasst werden, ist die auf dem Server gespeicherte Position nach jeder Positionsänderung des Mobilgeräts nicht mehr aktuell. Um die Position auf dem Server aktuell zu halten, gibt es mehrere Ansätze. Hier werden generell die beiden Techniken des *Querying Protocols* und des *Reporting Protocols* unterschieden [33]. Während bei einem Querying Protocol der Server die Clients aktiv nach ihren Positionen fragt, senden die Clients beim Reporting Protocol diese Informationen automatisch an den Server.

Da auch im Kontext dieser Masterarbeit die Datenerfassung der Mobilgeräte gezielt gesteuert werden soll, muss der Server nicht nur wissen, welche Geräte momentan angeschaltet sind, er braucht außerdem Informationen darüber, wo sich diese Geräte momentan befinden, um ihnen standortabhängige Anweisungen geben zu können. Eine einfache Lösung, bei der jedes passive Mobilgerät jede Positionsänderung in Echtzeit an den Server übermittelt, würde zwar eine hohe Aktualität der Positionsinformationen auf dem Server garantieren, aber auch zu hohen Sensor- und Kommunikationskosten führen. Da die Schonung von Ressourcen der Mobilgeräte im Vordergrund steht, scheidet diese Lösung aus. Alternativ sendet ein passives Gerät, nach einem vorgegebenen Zeitintervall Δ_{UT}, ein Update seiner Position an den Server, welcher diese Position speichert. Die Wahl der Größe des Intervalls zwischen den Positionsaktualisierungen kann dabei unterschiedliche Effekte nach sich ziehen. Ein kleiner Wert hat den Vorteil, dass der Server über relativ aktuelle Positionsinformationen der passiven Geräte verfügt, er erhöht aber auch das Nachrichtenaufkommen im System und den Ressourcenverbrauch der Mobilgeräte, da zusätzliche Sensor- und Kommunikationskosten verursacht werden. Angesichts der Tatsache, dass die meisten Menschen, und somit auch die Mobilgeräte, nur relativ selten langfristige Veränderungen ihres gewohnten Aufenthaltsorts durchführen, kann Δ_{UT} relativ groß gewählt werden, ohne dass die Positionsinformationen des Server zu schnell zu ungenau werden.

Neben der bisher vorgestellten Methode, Aktualisierungen der Position periodisch an den

Server zu schicken (*time-based*), gibt es den Ansatz die Aktualisierung auf Basis der Distanz zum letzten übertragenen Punkt durchzuführen (*distance-based*), indem eine Schranke für die maximale zulässige Abweichung definiert wird [6], oder eine Mischung aus beiden Ansätzen (*spatio-temporal*) [17]. Beim Distance-based-Ansatz überprüft jedes passive Mobilgerät nach Δ_{UT} seine aktuelle Position, sendet diese aber nur an den Server, wenn die Abweichung zur letzten an den Server übermittelten Position eine maximal zulässige Abweichung Δ_{UD} übersteigt. Dadurch können die Kommunikationskosten, auf Kosten der Aktualität der Position beim Server, weiter gesenkt werden.

Diese Technik nutzen wir, um Informationen über die Positionen der Mobilgeräte vor dem eigentlichen Start der Kartenerstellung zu verwalten, da diese für die Initialisierung der Erfassung besonders wichtig sind (siehe Abschnitt 3.4.4). Wurde die Erfassung bereits eingeleitet, so senden die im Zielbereich befindlichen Mobilgeräte, wie wir in den nächsten Abschnitten sehen werden, in regelmäßigen Abständen Nachrichten an den Server, in denen ihre jeweilige Position enthalten ist. Damit wird ein explizites Lokationsmanagement für Geräte innerhalb des Zielbereichs überflüssig und beschränkt sich auf Geräte, die sich außerhalb von diesem befinden.

3.4.3 Unterteilung des Zielbereichs

Um die Datenerfassung möglichst ressourcenschonend durchführen zu können, unterteilen wir den Zielbereich mit Hilfe des in Abschnitt 3.1.1 eingeführten Gitternetzes. Der Zweck besteht darin, die Erfassung gezielt auf Zellen zu konzentrieren, die nach der in Abschnitt 3.3.3 definierten Qualitätsmetrik für Gitterzellen, einen schlechten Qualitätswert im Vergleich zu den anderen Zellen aufweisen. Dies bedeutet, dass Zellen, die einen überdurchschnittlich guten Qualitätswert haben, von der Erfassung ausgesetzt werden, um die Ressourcen der Mobilgeräte zu schonen. Dazu definieren wir für jede Gitterzelle einen Status, der entweder *aktiv* oder *passiv* sein kann. Ein aktiver Zellstatus veranlasst die Mobilgeräte in dieser Zelle, wie gewohnt GPS-Daten zu sammeln. Betritt ein Mobilgerät hingegen eine Zelle, deren Status passiv ist, so wird die Erfassung von GPS-Daten solange ausgesetzt, wie sich das Gerät in dieser Zelle befindet. Das Gerät ändert seinen Status ebenfalls auf passiv. Aktive Geräte befinden sich daher in aktiven Zellen, während sich passive Geräte in passiven Zellen, in geschlossenen Räumen (fehlendes GPS-Signal) oder außerhalb des Zielbereichs befinden. Welchen Status welche Zelle gerade hat, speichert der Server in einem Statusvektor. Wie genau der Server entscheidet, welche Zellen aktiv oder passiv sind, erläutert Abschnitt 3.4.7.

Nehmen wir an, dass die Gitterzelle $C_{n,m}$ hinsichtlich der in Abschnitt 3.3.3 definierten Qualitätsmetrik momentan überdurchschnittlich gut erfasst ist und der Server daher entscheidet, dass diese Zelle vorübergehend von der Erfassung ausgeschlossen werden soll.

Sie erhält den Status passiv. Um nun den Mobilgeräten diese Statusänderung bekannt zu machen, sendet der Server den aktualisierten Statusvektor der Zellen mit einem aktuellen Zeitstempel an alle aktiven Geräte, welches sich innerhalb dieser Zelle bzw. in einer angrenzenden Zelle befinden, da diese direkt von dieser Änderung betroffen sind bzw. bei einem Zellwechsel davon betroffen sein werden. Jedes Gerät, das diese Nachricht empfangen hat, ersetzt nun seinen bisherigen Statusvektor und überprüft, ob die Zelle in der es sich befindet laut dem neuen Statusvektor erfasst werden muss oder nicht. Erkennt das Gerät, dass es sich nun in einer passiven Zelle befindet, stellt es die Erfassung der GPS-Daten ein, sendet den gerade erfassten Track an den Server und meldet sich diesem als passiv. Um Energie zu sparen, schalten das Geräte daraufhin seinen GPS-Empfänger aus.

3.4.4 Initialisierung der Erfassung

Schaltet ein Teilnehmer sein Mobilgerät ein, so registriert sich dieses beim Server, welcher dessen Geräte-ID und die Adresse seiner Mobilfunkschnittstelle für die spätere Kommunikation speichert. Die Serveradresse ist dabei jedem Mobilgerät a priori als bekannt vorausgesetzt. Wurde dem Server bereits ein Befehl zur Erstellung einer Karte erteilt, sendet er diesem Gerät eine Initialisierungsnachricht. Diese Initialisierungsnachricht enthält die Koordinaten des Zielbereichs, die Größe der Gitterzellen, den Status der einzelnen Gitterzellen, sowie alle anderen wichtigen Systemparameter, die im weiteren Verlauf dieses Kapitels eingeführt werden. Nachdem das Mobilgerät diese Nachricht erhalten hat, überprüft es, ob es sich innerhalb einer aktiven Zellen befindet. Ist dies der Fall, teilt es dies dem Server mit, setzt seinen Status auf aktiv und beginnt mit der Erfassung von GPS-Daten, andernfalls bleibt der Status des Geräts auf passiv. Ging beim Server zu diesem Zeitpunkt noch nicht der Befehl zur Erstellung einer Karte ein, so speichert dieser lediglich die Geräte-ID des Mobilgeräts und dessen letzte Position.

Sobald der Server einen Auftrag zur Kartografie eines Bereich erhält, kontaktiert er jedes Mobilgeräte, das sich laut seiner letzten auf dem Server gespeicherten Positionsinformation in der Nähe des Zielbereichs oder darin befindet, um diesen den Start der Erfassung durch das Senden einer Initialisierungsnachricht mitzuteilen. Erhält ein Mobilgerät diese Nachricht, führt es eine GPS-Ortung durch und überprüft anhand seiner Position, ob es sich im Zielbereich befindet. Ist dies der Fall, meldet es sich dem Server als aktiv und beginnt mit der Datenerfassung. Da zu Beginn der Zellstatus aller Zellen aktiv ist, müssen die Geräte lediglich überprüfen, ob sie sich innerhalb des Zielbereichs befinden oder nicht. Welche Geräte sich vor dem Start der Erfassung in der Nähe des Zielbereichs befinden, ermittelt der Server mit Hilfe des in Abschnitt 3.4.2 beschriebenen Lokationsmanagements. Befindet sich kein Gerät innerhalb oder nahe des Zielbereichs, führt der Server einen Broadcast der Initialisierungsnachricht an alle ihm bekannten Geräte durch, worauf alle Geräte, die sich

in der Nähe dieses Bereichs befinden, mit ihrer aktuellen Position antworten. Dies wiederholt er in regelmäßigen Zeitabständen solange bis sich mindestens ein Gerät als aktiv zurückmeldet.

Da möglichst alle Mobilgeräte im Zielbereich für die Erfassung aktiviert werden sollen, die Positionsinformationen der Mobilgeräte auf dem Server aber unter Umständen veraltet sind und der Server daher die tatsächliche Menge der sich zurzeit im Zielbereich befindlichen Mobilgeräte nicht kennt, muss er auf einem anderen Weg versuchen sicherzustellen, dass alle Geräte im Zielbereich die Initialisierungsnachricht erhalten. Eine einfache Lösung bestände darin, einen Mobilfunk-Geocast ins Zielgebiet zu senden, welcher allen dort befindlichen Mobilgeräten die Initialisierungsnachricht übermittelt. Dazu würden nur die Mobilfunksendemasten die Broadcast-Nachricht in ihre assoziierten Zellen schicken, wenn eine dieser Zellen Teil des Zielbereichs ist. Das Problem bei dieser Lösung liegt in den Restriktionen der Mobilfunktechnik. Zwar gibt es vereinzelte Lösungen, um eine Geocast über Mobilfunktechnik zu ermöglichen, GSM bietet zum Beispiel einen *Cell Broadcast* an, jedoch sind diese nicht ohne weiteres nutzbar und den Mobilfunkprovidern vorbehalten. Wie in Abschnitt 2.2.3 erwähnt, kann ein Geocast auch über ein mobiles Ad-hoc-Netzwerk gesendet werden. Dies nutzt der Server, um den Geräten, welche sich nach Senden der Initialisierungsnachricht als aktiv zurückgemeldet haben, den Auftrag zu erteilen einen Ad-hoc-Geocast mit der Initialisierungsnachricht in das Zielgebiet zu schicken. Dadurch werden auch Mobilgeräte erreicht, welche aufgrund veralteter Positionsinformationen zuvor nicht vom Server kontaktiert wurden, da ihre gespeicherte Position auf dem Server noch weit außerhalb des Zielbereichs war. Zur Veranschaulichung sind die Vorgänge auf dem Server und den Mobilgeräten in Abbildung 3.16 in einem Ablaufdiagramm dargestellt.

3.4.5 Übertragung der GPS-Daten

Haben die Geräte mit der Erfassung begonnen, müssen sie die gesammelten Daten zeitgemäß zum Server übertragen werden, um eine gewisse Aktualität des Straßenmodells zu gewährleisten und die Größe der zu übertragenden Nachrichten zu begrenzen. Auf der anderen Seite dürfen die erfassten Daten nicht zu häufig gesendet werden, da sonst das hohe Nachrichtenaufkommen den Nutzen der Datenaktualität aufhebt. Wir legen daher fest, dass jedes Gerät seinen gerade erfassten Track automatisch an den Server überträgt, nachdem es p Wegpunkte erfasst hat. Der Wert p wird dabei für jedes Gerät maximal so groß gewählt, dass die zwischengespeicherten Wegpunkte in den Speicher des Geräts passen und somit ein Speicherüberlauf ausgeschlossen ist. Wie groß p letztendlich gewählt wird, wird vom Systems festgelegt, indem der Wert für p in die Initialisierungsnachricht codiert wird, welche jedes Gerät bei Beginn der Erfassung erhält. Ein kleines p garantiert eine hohe Aktualität des Straßenmodells auf dem Server, verursacht aber auch ein hohes Nach-

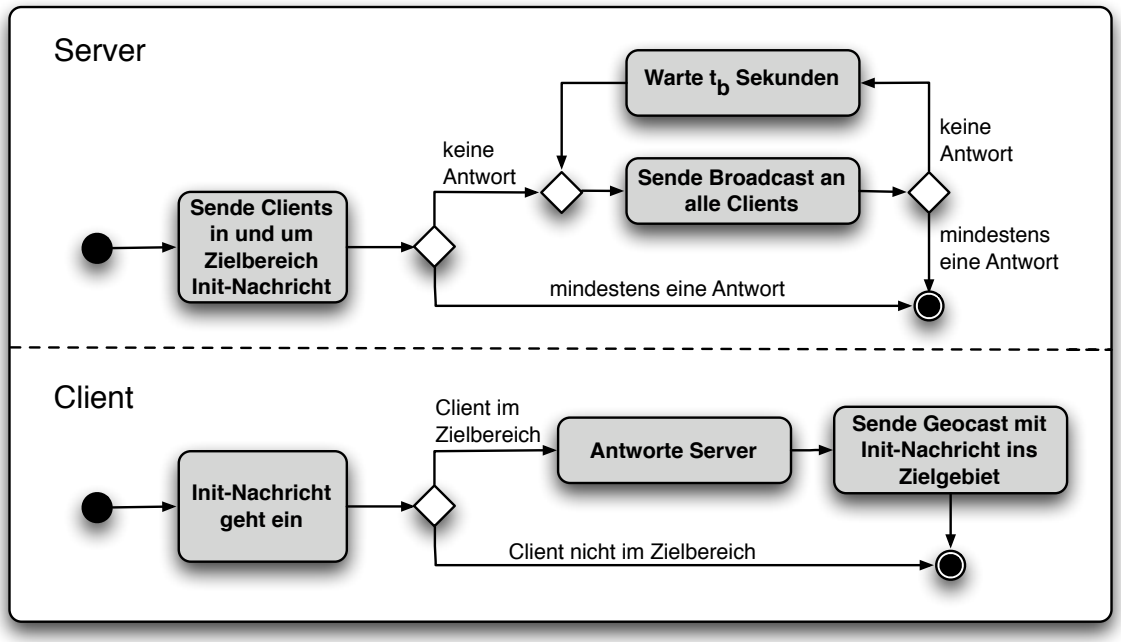


Abbildung 3.16: Ablauf der Initialisierung

richtenaufkommen und verbraucht somit mehr Ressourcen pro Mobilgerät. Ein zu großer Wert für p schont zwar die Ressourcen, verzögert aber die Übertragung der Wegpunkte zum Server und bewirkt somit, dass das Straßenmodell des Servers nicht den tatsächlichen Stand der Erfassung widerspiegelt und der Server daher seine Entscheidungen über den Fortgang der Erfassung anhand veralteter Daten trifft.

Erkennt ein Mobilgeräte durch eine GPS-Ortung, dass es sich außerhalb des Zielbereichs oder in einer passiven Zelle befindet und somit die Erfassung von GPS-Daten stoppt, so sendet es seinen aktuellen Track direkt zum Server, auch wenn es noch keine p Wegpunkte erfasst hat und teilt diesem mit, dass sein Status passiv ist.

Da ein Mobilgerät, nachdem es p Wegpunkt erfasst hat, diese an den Server schickt, obwohl es eventuell auf der gleichen Straße weitere Wegpunkte erfasst, muss der Server versuchen, um eine Zerstückelung von Straßen zu vermeiden, zusammengehörige Tracks wieder zu einer Straße zusammenzufügen. Dazu vergleicht er für jeden eingehenden Track, bevor das Fusionsverfahren nach Roth aus Abschnitt 3.2.2 durchgeführt wird, dessen ersten bzw. letzten Wegpunkt mit den ersten bzw. letzten Wegpunkten aller bereits im Straßenmodell vorhandenen Straßen. Er berechnet für diese jeweils deren Abstand d und fügt, falls dieser Abstand hinreichend gering ist, den neuen Track zur passenden Straße hinzu (siehe Abbildung 3.17).

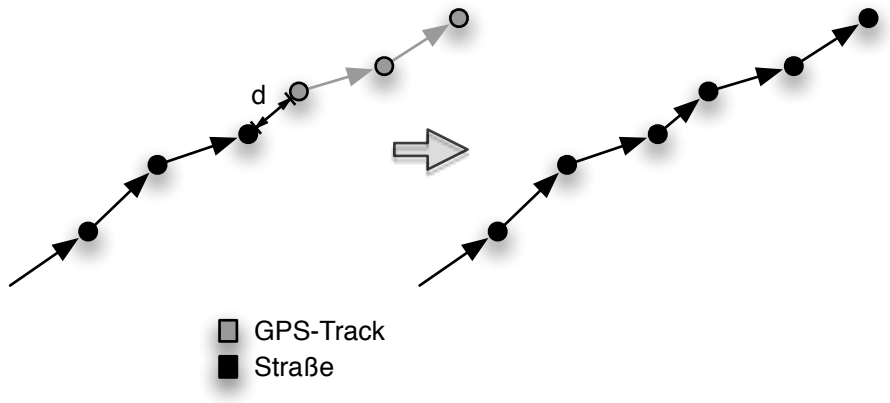


Abbildung 3.17: Verbindung eines Tracks mit einer Straße

3.4.6 Erkennung von neuen Geräten

Wird ein Gerät im Zielbereich eingeschaltet, so registriert sich dieses automatisch beim Server, um von diesem die nötigen Erfassungsinformationen zu erhalten. Anders verhält es sich mit passiven Geräten, die sich frisch in den Zielbereich oder von einer passiven in eine aktive Zelle bewegen. Da ihre GPS-Empfänger ausgeschaltet sind, können sie nicht feststellen, dass sie sich in einer aktiven Zelle befinden und eigentlich damit beginnen sollten GPS-Daten zu erfassen. Um dieses Problem zu lösen, werden nachfolgend zwei Verfahren vorgestellt, welche unterschiedliche Lösungen für dieses Problem bieten.

Ad-hoc-Ansatz

Um passiven Geräten mitzuteilen, dass sie sich in einer aktiven Zelle befinden, versenden die Mobilgeräte, welche sich in einer aktiven Zelle befinden, in regelmäßigen Abständen Initialisierungsnachrichten über ihre Ad-hoc-Schnittstelle an ihre unmittelbare Umgebung. Empfängt ein passives Gerät eine solche Initialisierungsnachricht, so überprüft es seine aktuelle Position via GPS. Wenn es sich in einer aktiven Zelle befindet, meldet es sich beim Server als aktiv, beginnt mit der Datenerfassung und sendet von nun selbst Ad-hoc-Nachrichten aus.

Diese Lösung verursacht jedoch unter Umständen ein relativ hohes Nachrichtenaufkommen. Speziell im Zentrum einer aktiven Zelle kann angenommen werden, dass fast alle Geräte aktiv sind, da sie aufgrund ihres Abstands zum Rand der Zelle schon länger darin verweilen oder dort eingeschaltet wurden. In diesem Gebiet Ad-hoc-Nachrichten zu versenden wäre daher mit hoher Wahrscheinlichkeit redundant, da ein Mobilgerät im ersten Fall mit hoher

Wahrscheinlichkeit schon eine Ah-hoc-Nachricht gehört hat oder beim Einschalten automatisch seine Position überprüft. Anders hingegen sieht es im Randbereich der Zelle aus, da hinzugekommene Geräte sich zuerst darin befinden werden. Um diese Eigenheit nutzen zu können, definieren wir einen inneren Randbereich für Gitterzellen und beschränken das Senden von Ad-hoc-Nachrichten auf Mobilgeräte, die sich in diesem Randbereich befinden (siehe Abbildung 3.18). Die Breite dieses Bereichs wird in die Initialisierungsnachricht codiert, um sie jedem Gerät bekannt zu machen.

Aktive Geräte, die sich nach Definition in einer aktiven Zelle befinden, überprüfen nach einem Zeitintervall von t_{adhoc}, ob sie sich innerhalb des Randbereichs ihrer momentan befindlichen Zelle befinden. Ist dies der Fall, senden sie eine Ad-hoch-Nachricht an ihre Umgebung aus, welche eine aktuelle Version der Initialisierungsnachricht enthält. Empfängt ein passives Gerät diese Nachricht, bestimmt es seine aktuelle Position. Stellt es fest, dass es sich ebenfalls in einer aktiven Zelle befindet, ändert es seinen Status auf aktiv und beginnt mit Erfassung der GPS-Daten. Befindet es sich noch außerhalb der aktiven Zelle, reagiert es nicht. Da das passive Gerät mit dieser Nachricht die Initialisierungsnachricht empfängt prüft es, ob der Zeitstempel dieser Nachricht neuer ist, als der der gespeicherten Initialisierungsnachricht. Ist der empfangene Zeitstempel neuer, aktualisiert es diese entsprechend. Dieser Fall kann eintreten, wenn der Server den Statusvektor aktualisiert, da dieser nur an aktive Geräte geschickt wird (siehe Abschnitt 3.4.3).

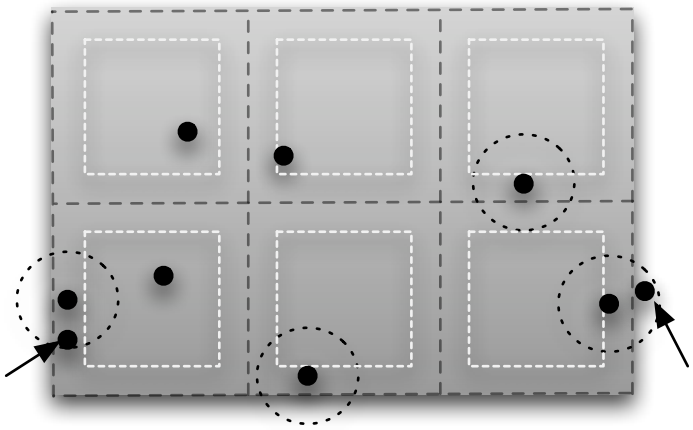


Abbildung 3.18: Definition eines Randbereichs für einzelne Zellen

Bei der Wahl der Größe des Randbereichs sollte außerdem die Reichweite von Ad-hoc-Verbindungen (siehe Abschnitt 2.2.3) im Verhältnis zum Zielbereich berücksichtig werden. Ist der Zielbereich um ein Vielfaches größer als diese Reichweite, sollte der Randbereich größer gewählt werden, um die Chancen der Mobilgeräte zu erhöhen, die Ad-hoc-Nachrichten

anderer Geräte zu hören. Durch den Randbereich wird die Anzahl der insgesamt versendeten Ad-hoc-Nachrichten reduziert, wobei die Wahrscheinlichkeit steigt, dass ein passives Gerät keine Initialisierungsnachricht erhält, obwohl es sich innerhalb einer aktiven Zelle befindet.

Da es vorkommen kann, dass ein passives Gerät eine aktive Zelle betritt, in der sich momentan keine anderes aktives Gerät befindet, erhält es keine Ad-hoc-Nachricht und wird daher nicht darüber informiert, dass es sich in einer aktiven Zelle befindet. Um diesen Sonderfall einzuschränken, definieren wir zusätzlich einen Grenzwert durch die Zeitschranke t_{thre}. Befindet sich ein Gerät in einer passiven Zelle und hat es nach Ablauf von t_{thre} noch keine Ad-hoc-Nachricht empfangen, so führt es eine GPS-Ortung durch und überprüft seine Position. Dabei sollte jedoch t_{thre} um ein Vielfaches größer gewählt werden als t_{adhoc}, da die Ad-hoc-Kommunikation als primäres Kommunikationsmittel genutzt werden soll und dadurch die GPS-Ortung vermieden werden soll. Für die später folgenden Implementierung, legen wir diesen Wert standardgemäß auf das Dreifache des Wertes von t_{adhoc} fest. Einen Ansatz bei dem auf Ad-hoc-Nachrichten gänzlich verzichtet wird, wird nachstehend vorgestellt.

Zeitbasierter Ansatz

Obwohl wir im Systemmodell von Mobilgeräten ausgehen, welche dazu fähig sind Ad-hoc-Nachrichten zu versenden und zu empfangen, wird diese Möglichkeit durch die Mehrzahl der Geräte-APIs noch nicht unterstützt und muss daher aufwändig selbst implementiert werden. Daher bietet sich für Geräte, die die Ad-hoc-Technologie nicht ohne weiteres unterstützen, folgender alternativer Ansatz:

Jedes passive Mobilgerät prüft nach t_w Sekunden seine aktuelle Position. Befindet es sich nicht mehr in der passiven Zelle und betritt eine aktive, meldet es sich beim Server wieder als aktiv an. Stellt das Gerät fest, dass es sich noch in der passiven Zelle befindet, so bleibt es weiterhin blockiert und führt die nächste Positionsbestimmung in t_w Sekunden durch. Im einfachsten Fall würde ein passives Gerät versuchen t_w so gering wie möglich zu wählen, da es dann den Wechsel in eine aktive Zelle umgehend bemerken würde. Dies würde jedoch eine permanente GPS-Ortung voraussetzen und somit die Ressourcen genauso stark belasten, wie wenn das Gerät aktiv wäre. Daher muss das Gerät versuchen t_w möglichst groß zu wählen, dabei aber trotzdem das Verlassen der blockierten Zelle frühestmöglich zu erkennen. Eine weit verbreitete Technik besteht darin, t_w hinsichtlich der aktuellen Geschwindigkeit (die aus der letzten GPS-Messung hervorgeht) und dem Abstand zum Rand der Gitterzelle zu wählen [17]. Daraus ergibt sich ein Wert für t_w, der (bei konstanter Geschwindigkeit) die Zeit bis zum Erreichen des Zellenrands angibt, wenn sich der Sensoren auf direktem Weg darauf zu bewegen würde. Das Mobilgerät würde daher einen Zellen-

wechsel rechtzeitig erkennen. Erhöht das Mobilgerät seine Geschwindigkeit, wird es durch diese Metrik den Zellenwechsel zu spät erkennen. Kennt ein Gerät daher seine maximale Geschwindigkeit, so wählt es diese anstelle der aktuellen Geschwindigkeit. Dadurch ist zwar die Wahrscheinlichkeit höher, dass das Gerät seine Position überprüft, wenn es sich noch innerhalb der Zelle befindet, ein zu spätes Erkennen des Wechseln der Zelle ist dadurch aber ausgeschlossen.

Für die später folgende Implementierung und der darauf folgenden Evaluation nehmen wir an, dass die Mobilgeräte die Fähigkeit besitzen Ad-hoc-Nachrichten zu senden und zu empfangen. Wir gehen daher standardmäßig von der Verwendung des Ad-hoc-Ansatzes für den Rest dieser Arbeit aus.

3.4.7 Statusänderungen bei Gitterzellen

Wie bereits in Abschnitt 3.4.3 erwähnt, kann die Erfassung der Straßen des Zielbereichs auf einzelne Gitterzellen beschränkt werden, um somit Ressourcen einzusparen. Der Server muss daher entscheiden, welche Zellen erfasst werden sollen und welche, aus Gründen der Ressourcenschonung, vernachlässigt werden können. Als Entscheidungsgrundlage nutzt er hierfür die in Abschnitt 3.3.3 definierten Qualitätsmetriken für einzelne Gitterzellen, indem er direkt die zugehörigen Werte der einzelnen Zelle vergleicht.

Dazu wird zuerst der Mittelwert der Qualität über alle Zellen gebildet und dann, mit Hilfe dieses Durchschnitts und eines übergebenen Deltas, über den Status jeder Zelle entschieden. Ist der Qualitätswert einer Zelle um ein Delta größer als der Durchschnitt der Qualität aller Zellen, erhält diese Zelle den Status passiv, wenn nicht den Status aktiv. Das Delta bleibt über die gesamte Systemlaufzeit konstant und wird zu Beginn vom Systembenutzer als Parameter übergeben. Je größer der Benutzer Delta wählt, desto weniger Zellen erhalten einen passiven Status. Dadurch werden mehr Daten erfasst, jedoch steigt auch gleichzeitig der Ressourcenverbrauch. Die Wahl eines kleinen Deltas bewirkt genau den umgekehrten Effekt. Eine genaue Beschreibung dieser Statusentscheidung zeigt Algorithmus 1 in Pseudocodenotation.

Algorithmus 1 wird jedes Mal aufgerufen, wenn sich mindestens ein Qualitätswert einer Gitterzellen ändert. Dies kann durch Hinzufügen einer neuen Straße zum Straßenmodell oder durch das Update einer bereits vorhanden Straße geschehen (siehe Abschnitt 3.2.2). Ändert sich der Status mindestens einer Zelle, sendet der Server den aktualisierten Status aller Gitterzellen mit einem aktuellen Zeitstempel an alle aktiven Mobilgeräte, welche sich in einer Zelle, deren Status geändert wurde, oder einer Nachbarzelle einer solchen Zelle, befinden. Diese prüfen daraufhin, ob sich der Status ihrer aktuellen Zelle geändert hat und reagieren entsprechend darauf.

Algorithmus 1 UPDATECELLSTATUS(*delta*)

1: $sum \leftarrow 0$
2: $meanValue \leftarrow 0$
3: **for** $\forall c \in$ Cells **do**
4: $\quad sum \leftarrow sum + c.getQuality()$
5: **end for**
6: $meanValue \leftarrow sum/|Cells|$
7: **for** $\forall c \in$ Cells **do**
8: $\quad$ **if** $c.getQuality > (meanValue + delta)$ **then**
9: $\quad\quad c.status \leftarrow passive$
10: $\quad$ **else**
11: $\quad\quad c.status \leftarrow active$
12: $\quad$ **end if**
13: **end for**

Anhand dieses Algorithmus, kann man sich nun überlegen, wie sich die Erfassung mit der in Abschnitt 3.3.3 definierten Qualitätsmetrik für Gitterzellen verhält, die in den Zeilen 4 und 8 ausgelesen wird. Dabei spielt der in der Qualitätsmetrik integrierte Zeitwert eine wichtige Rolle, wie wir im nachfolgenden Beispielszenario sehen werden.

3.4.8 Beispielszenario

Um deutlich zu machen, wie sich der in diesem Abschnitt eingeführte Erfassungsalgorithmus bezüglich den zuvor definierten Qualitätsmetriken verhält, wollen wir an dieser Stelle ein Beispielszenario erläutern, welches dieses Zusammenspiel veranschaulicht.

Um uns dabei auf das Wesentliche konzentrieren zu können, betrachten wir einen stark vereinfachten Erfassungsbereichs, der nur aus zwei Gitterzellen besteht. Die erste Gitterzelle beinhaltet sowohl Straßen, die von den Mobilgeräten sehr stark frequentiert werden, als auch Straßen, die nur selten besucht werden. Die zweite Gitterzelle enthält nur wenige Straßen, die selten besucht werden. Das im vorherigen Abschnitt eingeführte Delta, welches für die Entscheidung, welche Zelle eine passiven Status erhalten sollen, nötig ist, belegen wir mit einem Wert von 0.1.

Nach Beginn der Erfassung werden die oft besuchten Straßen aus der ersten Gitterzelle schnell einen hohen Qualitätswert erhalten, da dieser mit jeder Interpolation der auf dieser Straße erfassten Tracks steigt. Nehmen wir an, der Durchschnittswert dieser Straßen wäre momentan bei 0.5. Der durchschnittliche Qualitätswert der Straßen der zweiten Gitterzelle sei hingegen eher gering bei 0.1. Würden wir jetzt den Zeitwert als Qualitätsmetrik nicht berücksichtigen, wäre der durchschnittliche Qualitätswert der beiden Zellen bei 0.3 und der Wert der ersten Zelle um 0.2 höher als dieser Durchschnitt. Die Zelle würde somit

einen passiven Status erhalten, obwohl die Erfassung erst begonnen hat und die meisten Straßen darin noch nicht entdeckt wurden. Hier kommt die Funktion des Zeitwerts ins Spiel. Da der Zeitwert zu Beginn noch relativ gering ist, wir nehmen einen Wert von 0.1 an, dämpft er die Qualitätswerte der Gitterzellen auf 0.05 bzw. 0.01. Der durchschnittliche Qualitätswert aller Zellen liegt jetzt nur noch bei 0.03, der Qualitätswert der ersten Zelle ist nur noch um 0.02 höher als dieser Wert. Die Zelle erhält also keinen passiven Status. Unter der Annahme, dass die Qualitätswerte der Zellen konstant bleiben, würde die erste Zelle erst dann den Status auf passiv wechseln, wenn der Zeitwert einen Wert von 0.5 erreicht hat. Bis dahin erhöht sich die Chance, dass die bisher unentdeckten Straßen in der ersten Zellen ebenfalls erfasst werden und somit nicht unentdeckt bleiben, wie es der Fall gewesen wäre, wenn die Zelle schon zu Beginn einen passiven Status erhalten hätte und somit die Erfassung für diese Zelle unterbunden gewesen wäre.

Dieses Beispiel hat die Wichtigkeit des Zweitwerts der Erfassung hervorgehoben, der die Chancen erhöht, dass auch weniger oft besuchte Straßen erfasst werden. In den folgenden Abschnitten, wollen wir einige optionale Überlegungen anstellen, welche als Erweiterung des bisher vorgestellten Erfassungsalgorithmus gesehen werden können.

3.4.9 Kategorisierung von Straßen

Neben den Koordinaten der aktuellen Position und der Varianz der Messung, enthält jede GPS-Messung außerdem die aktuelle Geschwindigkeit des Sensors zum Zeitpunkt der Erfassung. Dieser Wert kann genutzt werden, um Rückschlüsse über die Art der Straße zu ziehen, welche der Sensor passiert hat. Während reine Fußgängerwege selten mit einer Geschwindigkeit von mehr als wenigen km/h passiert werden, können Kraftfahrzeugstraßen eine Vielzahl von verschiedenen Kategorien annehmen. Um eine Klassifikation dieser zu ermöglichen, unterteilen wir die Straßen in Kategorien, welche in Anlehnung an die in Deutschland vorherrschenden Tempolimits definiert werden.

Straßenklassen

Gemäß der deutschen Straßenverkehrsordnung existieren auf Straßenabschnitten Tempolimits, welche sich grob in die in Tabelle 3.1 selbstdefinierten Klassen einteilen lassen:

Während sich die Kategorien A bis F auf Fahrzeuge beziehen, werden die Mobilgeräte zusätzlich von Fußgängern und Fahrradfahrern mitgeführt, welche sich bedeutend langsamer fortbewegen. Da Fahrradfahrer sich hinsichtlich ihrer Geschwindigkeit kaum von Straßen der Kategorie A unterscheiden lassen, ist die Erkennung eines Fahrradweges nahezu unmöglich. Fußgängerwege können hingegen abgrenzt werden, da die dort erfasste Geschwindigkeit selten mehr als 7 km/h beträgt.

Tempolimit	Grund	Kategorie
30	ausgeschildert	A
50	innerorts	B
70	ausgeschildert	C
100	außerorts	D
120	ausgeschildert	E
-	Autobahn	F
7	Fußgängerweg	P

Tabelle 3.1: Straßenklassen

Anpassung des Fusionsverfahrens

Um eine Klassifizierung der erfassten Straßen zu ermöglichen, müssen möglichst viele Informationen über die Geschwindigkeiten vorliegen, mit welcher die Sensoren die jeweiligen Straßenpositionen passiert haben. Dazu müssen die Geschwindigkeitswerte bei der Verarbeitung der erfassten Tracks, möglichst ohne Verlust von Informationen, im Straßenmodell gespeichert werden. Das für die Fusion der Wegpunkte genutzte Verfahren zur Interpolation von zwei Punkten (siehe Abschnitt 3.2.2) erweist sich dabei als ungeeignet, da dadurch ein Informationsverlust eintritt, welcher die Kategorisierung der Straße erschwert. Das Verfahren wird daher folgendermaßen erweitert:

Werden zwei Straßenabschnitte wie in Abschnitt 3.2.2 beschrieben fusioniert, so werden wie bisher die Koordinaten und die Varianz des neuen Wegpunktes durch Interpolation der beiden alten berechnet. Hinsichtlich der Geschwindigkeiten wird dem neuen Wert jedoch nicht ein interpolierter Wert der beiden alten Punkte zugewiesen, sondern die beiden alten Werte werden zu einer Menge vereint und mit dem neuen Punkt gespeichert. Liegt für einen zu fusionierenden Punkt schon mehr als ein Geschwindigkeitswert in dieser Menge vor, so werden alle Geschwindigkeitswerte dieses Punkts in die Menge des neuen Punkts übernommen. Dadurch gehen keine Informationen verloren, da alle erfassten Geschwindigkeiten in dieser Menge gespeichert sind.

Klassifizierung

Mit Hilfe der Geschwindigkeitsangaben, die für jeden Punkt vorhanden sind, kann jede Kante einer Straße zu einer Kategorie aus Tabelle 3.1 zugeordnet werden. Dabei reicht es nicht aus, einfach den Durchschnitt aus den gemessenen Geschwindigkeiten der Knoten einer Kante zu berechnen, da viele Straßen sowohl von Autos, Fahrradfahrern als auch Fußgängern passierbar sind und daher keine eindeutige Aussage über das Geschwindigkeitslimit der Straße getroffen werden kann. Zudem kann es auf nahezu allen Straßenka-

tegorien zu Verkehrsbehinderungen oder Staus kommen. Dadurch sind die Verkehrsteilnehmer gezwungen teilweise erheblich langsamer als die für den Straßenabschnitt erlaubte Höchstgeschwindigkeit zu fahren, wodurch die Bestimmung der Straßenkategorie zusätzlich erschwert wird.

Um die Schwierigkeiten der Interpretation des Durchschnittswerts zu umgehen, ziehen wir für die Bestimmung der Straßenklasse den Maximalwert heran, der auf dieser Straße erfasst wurde. Dabei ist jedoch zu beachten, dass auch hier Werte ermittelt werden können, die über der eigentlichen Höchstgeschwindigkeit des Straßenabschnitts liegen, da sie von Fahrzeugen erfasst wurden, die mit überhöhter Geschwindigkeit unterwegs waren. Daher legen wir fest, dass wir für einen Wegpunkt jeweils die n höchsten Geschwindigkeiten des Wegpunkts zur Berechnung verwenden und aus diesen den Mittelwert bilden. Dabei muss bei der Wahl von n die Anzahl m der vorliegenden Geschwindigkeitswerte für den Wegpunkt berücksichtigt werden, wobei gilt das $n = \lceil c \cdot m \rceil$ mit $c \leq 1$. Welchen Wert c annehmen soll, kann letztendlich der Benutzer des System entscheiden. Als Standardwert wählen wir $c = 1/3$, welcher das obere Drittel der vorliegenden Geschwindigkeiten bei der Berechnung des Mittelwerts berücksichtigt.

Mithilfe des berechneten Mittelwerts, kann nun jeder Wegpunkt klassifiziert werden. Da der Mittelwert sowohl über, als auch unter, der tatsächlich zulässigen Geschwindigkeit der Straßenklasse liegen kann, wird jeder Wegpunkt der Kategorie zugeordnet, deren Tempolimit dem Mittelwert am nächsten liegt. Um nun eine Straßenkante zwischen zwei Wegpunkten zu klassifizieren, betrachten wir die zugeordneten Klassen der zugehörigen Wegpunkte der Kante. Sind diese gleich, so erhält die Kante die gleiche Kategorie wie die Wegpunkte. Sind diese jedoch unterschiedlich, so wählen wir einen pessimistischen Ansatz, der der Kante das Minimum der beiden Werte zuweist, da wir nicht wissen können, wo genau auf der Kante die Änderung des Tempolimits stattfindet.

Zuverlässigkeit

Aufgrund der schwer vorherzusagenden Dynamik des Straßenverkehrs, ist eine wirklich zuverlässige Klassifikation von Straßen nur schwer möglich. So wird speziell in Bereichen in denen der Verkehr sehr dicht ist, zum Beispiel in Städten, oder die Straßenführung sehr eng, kaum jemand die zulässige Höchstgeschwindigkeit fahren können. Daher kann es relativ häufig vorkommen, dass Straßen langsamer klassifiziert werden, als sie in Wirklichkeit sind. Auf der anderen Seite gibt das Verfahren jedoch eine relativ genau Abschätzung nach oben ab. So wird, vorausgesetzt die Mehrheit der Autofahrer hält sich an die bestehenden Tempolimits, eine Straße in den meisten Fällen nicht schneller klassifiziert als sie ist.

Im folgenden Abschnitt werden Vorschläge erbracht, wie die Klassifizierung der Straßen unter Mithilfe des Benutzers des Mobilgeräts weiter verbessert werden kann und somit die

Zuverlässigkeit der Straßenklassifizierung steigt.

3.4.10 Optimierungspotentiale

Die bisher vorgestellten Konzepte bilden die Grundlage des Erfassungsalgorithmus, der in sich vollständig ist und daher zur Erfassung von Straßenkarten genutzt werden kann. In diesem Abschnitt werden, auf dem bisherigen Ansatz aufbauende, optionale Konzepte vorgestellt, welche Verbesserungen für das System in Sinne von Ressourcenschonung und Erhöhung der Qualität der erfassten Straßen mit sich bringen.

Partizipation des Nutzers

Der bisher verfolgte Ansatz basiert auf einer rein opportunistischen Sensordatenerfassung, welche keine Interaktion der Mobilgerätebesitzer benötigt. Weicht man von dieser Annahme ab, bieten sich weitere Optimierungspotentiale, welche eine Verbesserung der Erfassung ermöglich.

Beim bisher verfolgten opportunistischen Ansatz, hat ein Teilnehmer keinerlei Informationen darüber, wie der aktuelle Zustand des Straßenmodells auf dem Server aussieht. Würde er diesen kennen, könnte er einsehen, welche Straßen bisher noch gar nicht bzw. nur relativ schlecht erfasst wurden. Diese könnte er dann aus eigener Motivation heraus gezielt entlanggehen und so die Erfassung des Systems beschleunigen bzw. die Qualität bereits vorhandener Bereiche verbessern. Projekte wie OpenStreetMap haben gezeigt, dass eine generelle Bereitschaft von Teilnehmer zu derartigen Ansätzen besteht.

Die Umsetzung könnte folgendermaßen aussehen: Die Anwendung auf dem Mobilgerät verfügt über eine Landschaftskarte, über welches sich ein Overlay zeichnen lässt. Auf Wunsch kann ein Teilnehmer vom Server das aktuelle Straßenmodell anfordern, welches Angaben zur Qualität der bisher erfassten Straßen, enthält. Diese werden auf der Karte des Mobilgeräts visualisiert, so dass der Teilnehmer sich ein Bild der aktuellen Erfassung machen kann. Ist das Straßenmodell bereits relativ groß und daher ungeeignet für eine Übertragung an das Mobilgerät, kann der Server die Übertragung auf Straßen beschränken, die sich im näheren Umfeld des Mobilgeräts befinden.

Neben der gezielten Erfassung von Straßen, kann der Benutzer, wie in Abschnitt 3.4.9 bereits erwähnt, die Qualität bei der Klassifizierung von Straßen zusätzlich erhöhen. So könnte der Benutzer, bevor ein Track an den Server gesendet wird, diesen selbst anhand der verfügbaren Straßenklassen klassifizieren.

Stillstand des Mobilgeräts

Da die Mobilgeräte von Menschen oder in Fahrzeugen mitgeführt werden, haben sie die unvermeidliche Eigenschaft, zu gewissen Zeitpunkten auf der gleichen Stelle zu verweilen. Sei es, weil ein Fußgänger oder ein Fahrzeug an einer Ampel wartet, oder weil das Gerät unbenutzt irgendwo liegt, eine ständige GPS-Ortung würde immer die gleiche Position ermitteln und somit unnötig Ressourcen verschwenden.

Verfügt das Mobilgerät über einen Beschleunigungssensor und kann somit unterscheiden, ob es sich bewegt oder ob es gerade steht, kann es diesen Nutzen um die GPS-Ortungen für die Zeit in der das Gerät stillsteht gezielt zu unterdrücken. Ein Mobilgerät überprüft daher nach jeder GPS-Ortung, ob es sich in der Zeit bis zur nächsten GPS-Ortung bewegt oder nicht. Hat es sich bewegt, führt es wie gewohnt die nächste Ortung durch. Fand keine Bewegung statt, wird die Ortung ausgelassen. Diese wird solange unterdrückt, bis sich das Gerät wieder bewegt.

Bei dieser Technik ist jedoch eine ständige Überwachung des Bewegungssensor nötig, daher muss der Ressourcenverbrauch des Bewegungssensors erheblich geringer sein, als der Ressourcenverbrauch einer GPS-Ortung, da sich sonst die Ressourcenersparnis aufhebt.

Vereinfachung von GPS-Tracks

Da alle GPS-Tracks von den Mobilgeräten an den Server übertragen werden, entstehen auch an dieser Stelle Kosten im Sinne von Ressourcenverbrauch, die man durch einige Optimierungen reduzieren kann, wie man sich anhand von einem Beispiel leicht überlegen kann. Angenommen ein Mobilgerät bewegt sich auf direkter Linie von Punkt A nach Punkt B. Auf der Strecke zwischen A und B führt es jedoch mehrere GPS-Ortungen durch, welche es dem aktuellen Track hinzufügt. Da diese Zwischenpunkte genau auf der Geraden zwischen A und B liegen, sind sie redundant und vergrößern die Größe des GPS-Tracks. Da dieser an den Server gesendet werden muss, werden so unnötig Ressourcen verbraucht, ohne einen Mehrwert an Informationen zu erhalten.

Die Vereinfachung von GPS-Tracks wird momentan in verschiedenen Forschungsveröffentlichungen diskutiert [31]. Viele dieser Verfahren basieren auf dem folgenden, einfachen Ansatz (siehe zum Beispiel [35] und [23]):

Für einen GPS-Track mit den Wegpunkten $(s_0, \cdots, s_n)$ wird, beginnend mit s_0, für jeden Wegpunkt s_i getestet, ob es möglich ist von s_i zu s_y (mit $i < y$) eine Kante zu legen, bei der jeder dazwischenliegende Wegpunkte s_x (mit $i < x < y$), einen Abstand zu dieser Kante hat, der geringer ist, als ein zuvor festgelegtes ϵ. Werden für einen Wegpunkt s_i mehrere solcher Kanten gefunden, wird s_i mit dem Wegpunkt s_l verknüpft, der chronologisch gese-

hen am weitesten hinten im Track liegt. Alle zwischen s_i und s_l befindlichen Wegpunkte werden aus dem Track entfernt und das Verfahren beim Wegpunkt s_l fortgesetzt. Das Ergebnis dieses Verfahrens ist ein Track, der gegebenenfalls weniger Wegpunkte enthält, aber dessen Abweichung zum vorherigen Track nie größer wie ϵ ist.

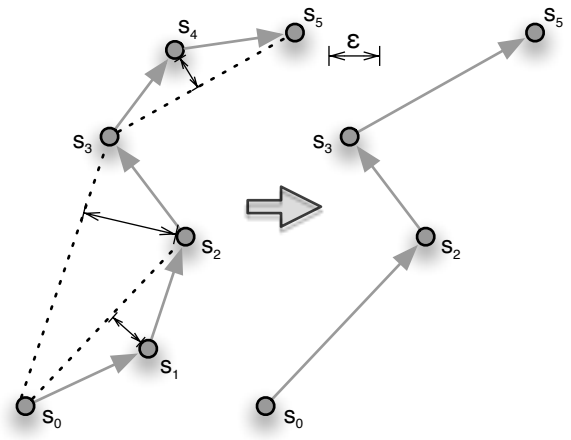


Abbildung 3.19: Vereinfachung eines GPS-Tracks

Abbildung 3.19 zeigt ein Beispiel für dieses Verfahren. Weil der Abstand von Punkt s_1 zur Kante (s_0, s_2) kleiner als ϵ ist, kann der Punkt s_1 zur Vereinfachung aus dem Track gestrichen werden. Da der Abstand von Punkt s_2 zur Kante (s_0, s_3) jedoch größer als ϵ ist, kann dieser Teil nicht weiter vereinfacht werden. Das Verfahren beginnt nun für Wegpunkt s_3 von neuem, wobei hier wiederum eine Vereinfachung zu Wegpunkt s_5 möglich ist.

Das Problem das dieses Verfahren mit sich bringt liegt darin, dass nicht nur Positionsinformationen von Wegpunkten verloren gehen, sondern auch zusätzliche Informationen, wie die Varianz oder die Geschwindigkeit. Da die Varianz aber als direkte Eingangsgröße für die in Abschnitt 3.3.1 definierte Qualitätsmetrik genutzt wird, müsste man die Informationen, die durch die wegfallenden Wegpunkte verloren gehen, bei der Vereinfachung berücksichtigen. So könnte man die bestehenden Verfahren so erweitern, dass eine Vereinfachung von Wegpunkt s_i zu Wegpunkt s_l nur erlaubt ist, wenn die Varianzen der dazwischenliegenden Wegpunkte nicht um mehr als ϵ_v von dem Durchschnitt der Varianzen von s_i und s_l abweichen. Da die Vereinfachung von Trajektorien nicht Ziel dieser Arbeit ist, wollen wir die Diskussion hier beenden und dieses Konzept als optional werten.

3.5 Zusammenfassung

Mit dem Abschluss dieses Kapitels wurden alle notwendigen Konzepte eingeführt, welche für das zu entwerfende System wichtig sind. Neben der Einführung in die theoretischen Grundlagen, die nötig sind um mehrere GPS-Tracks zu einer eindeutigen Straße zusammenzuführen, wurde eine Qualitätsmetrik vorgestellt, welche die erfassten Straßen hinsichtlich verschiedener Eigenschaften beurteilt. Mit Hilfe dieser Metrik konnte ein Erfassungsalgorithmus entworfen werden, der Entscheidungen darüber trifft, welche Bereiche zu Gunsten der Ressourcenschonung vernachlässigt werden können.

In der nachfolgenden Implementierung müssen diese Konzepte möglichst exakt in einer Programmiersprache abgebildet werden, um in der darauf folgenden Evaluation Rückschlüsse über verschiedene Testszenarios ziehen zu können. Dabei wurden die vorgestellten Konzepte bereits soweit detailliert, dass der folgende Abschnitt der Implementierung im Vergleich relativ kurz gehalten werden kann.

Kapitel 4

Implementierung

Nach dem Entwurf der Konzepte, können diese nun in einer konkreten Programmiersprache umgesetzt werden. Der folgende Abschnitt stellt diese Implementierung vor und erläutert die dazu verwendeten Werkzeuge und Implementierungsvarianten. Ziel der Implementierung ist es, ein System zu entwickeln, welches durch die in Kapitel 5 folgende Evaluation validiert und analysiert werden kann. Um dies bestmöglich zu gewährleisten, wurden die zuvor definierten Konzepte als Erweiterung des Netzwerksimulators ns-2 implementiert.

Aus diesem Grund stellt Abschnitt 4.1 den Netzwerksimulator ns-2, mit dessen zugehörigen Programmiersprachen, kurz näher vor. Danach beschreibt Abschnitt 4.2 in abstrakter Form, wie die Konzepte in diesen Programmiersprachen umgesetzt wurden und welche Konzepte bei der Implementierung keine Berücksichtigung fanden.

4.1 Der ns-2 Netzwerksimulator

Um für die, im nächsten Kapitel folgende, Evaluation eine Simulation des Systems durchführen zu können, ist ein Simulator nötig, der auf der einen Seite eine realistische Kommunikation der Mobilgeräte nachahmen kann und auf der anderen Seite die Fähigkeit besitzt, ein Szenario, bei dem sich mobile Knoten entlang virtueller Straßen bewegen, zu simulieren. Ein Simulator, der diese Anforderungen erfüllt, ist der Netzwerksimulator ns-2. Seine Eigenschaft, selbstgeschriebenen Programmcode integrieren zu können, macht ihn zudem beliebig erweiterbar und daher für die Bedürfnisse dieser Implementierung optimal. Um ein Grundverständnis für ns-2 zu bekommen, erläutert dieser Abschnitt einige Grundlagen, die bei Bedarf in der entsprechenden Literatur vertieft werden können (siehe zum Beispiel [3] und [5]).

Um einen Einstieg in die Funktionsweise des Simulators zu bekommen, bietet Abschnitt

4.1.1 eine kurze, allgemeine Einführung. Welche Rolle die beiden Programmiersprachen OTcL und C++ für den Simulator spielen und wie diese für die Simulationen verwendet werden, beschreiben die Kapitel 4.1.2 und 4.1.3.

4.1.1 Grundlagen

Ns-2 ist in der Forschungsgemeinde ein mittlerweile weit verbreiteter Netzwerksimulator, der es durch ein diskretes Eventmodell ermöglicht, Abläufe in einem Netzwerk zu simulieren. Der Simulator unterstützt eine große Auswahl an allen gängigen Übertragungsprotokollen, sowie verschiedene Netzwerktypen, darunter zum Beispiel auch ein Modus zur Simulation von Ad-hoc-Übertragungen. Um eine Simulation zu erstellen, legt der Benutzer Netzwerkknoten in Form von Objekten an, welche spezielle Charakteristiken, wie zum Beispiel die einer FTP-Anwendung, mit sich bringen. Um eine Kommunikation zwischen diesen Knoten zu ermöglichen, definiert der Benutzer Übertragungskanäle, welche diese Knoten untereinander verbinden. Diese können wiederum über eigene Eigenschaften, wie zum Beispiel über eine festgelegte Bandbereite, verfügen. Startet der Benutzer eine Simulation, so werden die Abläufe, die sich aus den Eigenschaften dieser Komponenten ergeben, durch den Simulator nachgeahmt. Diese reichen von der Simulation einer Paketkollision bis hin zu Delays bei der Übertragung. Bei der Simulation einer Ad-hoc-Kommunikation wird anhand der zuvor festgelegten Reichweite der jeweiligen Funkantenne und den Positionen der Knoten ermittelt, welche Knoten welche Datenpakete empfangen. Dabei werden charakteristische Probleme, wie das *Hidden Terminal Problem* [48], bei der Simulation berücksichtigt.

Um die Eigenheiten eines mobilen Szenarios simulieren zu können, werden so genannte Tracefiles unterstützt, die den mobilen Knoten Positionen auf virtuellen Straßen zuweisen und es somit ermöglichen, realistische Szenarios nachzustellen. Wie genau diese Tracefiles eingesetzt und erstellt werden, wird in der Evaluation in Abschnitt 5.1 näher erläutert.

Da eine detaillierte Beschreibung des Funktionsumfangs von ns-2 im Rahmen dieser Arbeit nicht möglich ist, werden wir im Folgenden nur kurz auf die beiden Programmiersprachen eingehen, die von ns-2 verwendet werden — OTcl und C++.

4.1.2 OTcl

OTcl ist eine objektorientierte Erweiterung der Skriptsprache Tcl (Tool Command Language). Sie dient im Rahmen von ns-2 als Schnittstelle zwischen den in C++ implementierten Objekten und dem Benutzer des Simulators.

Um eine Simulation zu erstellen, legt der Benutzer eine Skriptdatei im OTcl-Format an

und übergibt diese zur Ausführung dem ns-2 Simulator. In dieser Skriptdatei initialisiert der Benutzer alle für die Simulation benötigten Objekte und verknüpft diese entsprechend untereinander. Für die erstellten Objekte kann der Benutzer Methoden aufrufen und diesen Parameter übergeben, die zuvor in C++ definiert wurden. Für die zeitliche Steuerung der Ereignisse, verfügt ns-2 über einen Scheduler. Diesen kann der Benutzer anweisen, zu einem bestimmten Zeitpunkt, eine zuvor definierte Funktion aufzurufen.

Da es sich bei OTcl um eine Skriptsprache handelt, ist bei Änderungen in der Skriptdatei keine erneute Kompilierung des Simulators nötig (siehe Abbildung 4.1). Dadurch gestaltet sich eine Evaluation, bei der verschiedene Belegungen von Parametern getestet werden sollen, relativ einfach.

Abbildung 4.1: Interaktion des ns-2 Simulators

4.1.3 C++

Der Kern des ns-2 Simulators ist in der Programmiersprache C++ implementiert. In ihr sind alle Objekte und Methoden definiert, welche über die OTcl-Skriptdatei aufgerufen werden können. Somit ist auch die Möglichkeit gegeben, eigene Objekte in C++ zu schreiben und über die Skriptdatei aufrufbar zu machen. Da C++ jedoch kompiliert werden muss, ist nach jeder Änderung im Code eine erneute Kompilierung des ganzen Simulators notwendig. Bei Start des ns-2 Simulators wird die OTcl-Skriptdatei interpretiert und Objekt- und Methodenaufrufe werden mit den jeweiligen Stellen im C++ Code verglichen. Wird das entsprechende Gegenstück gefunden, wird der C++ Code ausgeführt. Ist das Objekt oder die Methode dem Simulator nicht bekannt, wirft die Anwendung einen Fehler.

Ns-2 bringt bereits eine Implementierung der wichtigsten Kommunikationsprotokolle mit sich. Will der Benutzer eigene Objekte in ns-2 erstellen, kann er mit Hilfe von Vererbung die Methoden dieser vordefinierten Objekte auch in seinen eigenen Klassen nutzen und gegebenenfalls erweitern. Dabei bietet es sich an, die in ns-2 bereits vorhandene Basisklasse *Agent* zu überschreiben, da diese bereits die wichtigsten Methoden, wie das Senden und Empfangen von Paketen, enthält.

4.2 Umsetzung der Konzepte

Wie bereits erwähnt, wurden die in Kapitel 3 beschriebenen Konzepte in ns-2 und somit in C++ und OTcl implementiert. Um diese Implementierung möglichst anschaulich zu beschreiben, wird die Unified Modeling Language (UML) verwendet und auf expliziten Programmcode verzichtet. Da für diesen Abschnitt die Kenntnisse der grundlegenden Elemente der UML vorausgesetzt werden, sei an dieser Stelle auf einige Werke verwiesen, welche eine Einführung in selbige geben — [18],[27].

Wir beginnen in Abschnitt 4.2.1 mit einem allgemeinen Teil, welcher eine generelle Übersicht darüber gibt, welche Rolle ns-2 in Rahmen dieser Arbeit spielt und wie die grundlegenden Objekte und Abläufe des Konzepts darin umgesetzt wurden. Um einen Überblick über die Systemarchitektur zu erhalten, stellt Abschnitt 4.2.2 die verwendeten Klassen anhand eines Klassendiagramms dar und gibt Erläuterungen zu diesen. Das Zusammenspiel dieser Klassen wird im anschließenden Abschnitt 4.2.3 durch Sequenz- bzw. Aktivitätsdiagrammen verdeutlich, bevor Abschnitt 4.2.4 erläutert, welche Konzepte im Rahmen dieser Implementierung nicht umgesetzt wurden.

4.2.1 Allgemeines

Die beiden wesentlichen Bestandteile, die im Rahmen dieser Arbeit genutzt werden, sind zum einen die Mobilgeräte, welche GPS-Daten erfassen, und zum anderen der zentrale Server, welcher diese Daten verarbeitet. Für die Implementierung der Mobilgeräte wurde in ns-2 eine Klasse geschrieben, welche von der Basisklasse *Agent* ableitet. Dadurch sind die Methoden zur Kommunikation mit anderen Mobilgeräten bereits gegeben. Als Kommunikationskanal wurde in der OTcl-Datei eine Ad-hoc-Übertragung festgelegt, so dass eine realistische Simulation der Gerätekommunikation untereinander gewährleistet ist. Der Server benötigt keine Ad-hoc-Schnittstelle und wurde daher als einfache C++ Klasse geschrieben, ohne eine der von ns-2 bereitgestellten Send- oder Receive-Methoden zu verwenden. Die Kommunikation zwischen den Mobilgeräten und dem Server wurde durch das gegenseitige Aufrufen der entsprechenden Methoden abstrahiert. Laut Konzeptteil kommunizieren diese Einheiten über eine Mobilfunktechnik, wie zum Beispiel GPRS. Dies kann aber im Rahmen der Implementierung vernachlässigt werden, da die Eigenheiten dieser Kommunikationsart keine Auswirkungen auf die in dieser Arbeit zu evaluierenden Konzepte haben.

Die Agent- und Serverklassen arbeiten jeweils mit einer Reihe von Parametern, welche im Konzeptteil dieser Arbeit bereits eingeführt wurden. Diese sollen bei der später folgenden Evaluation mit verschiedenen Werten belegt und mit diesen anschließend simuliert werden. Dazu ist jeder dieser Parameter in der OTcl-Datei änderbar, so dass mehrere Belegungen der Parameter simuliert werden können, ohne dass jeweils eine erneute Kompilierung des

Quellcodes nötig ist. Eine genaue Übersicht über die zu belegenden Parameter, bietet Tabelle 5.2 in Abschnitt 5.2.1.

Als nächstes werfen wir einen Blick auf die Klassen, die implementiert wurden, um die für das Konzept nötigen Funktionen umzusetzen.

4.2.2 Grundlegende Architektur

Die grundlegende Architektur des implementierten Systems besteht aus zehn Klassen, welche alle für das System nötigen Methoden und Variablen enthalten. Abbildung 4.2 zeigt diese Klassen, mit ihren jeweiligen Abhängigkeiten zueinander, in einem einfachen UML-Klassendiagramm. Hierbei wurde aus Platzgründen auf eine detailliertere Form des Klassendiagramms verzichtet und damit auch auf die explizite Aufführung von Methoden und Klassenattributen.

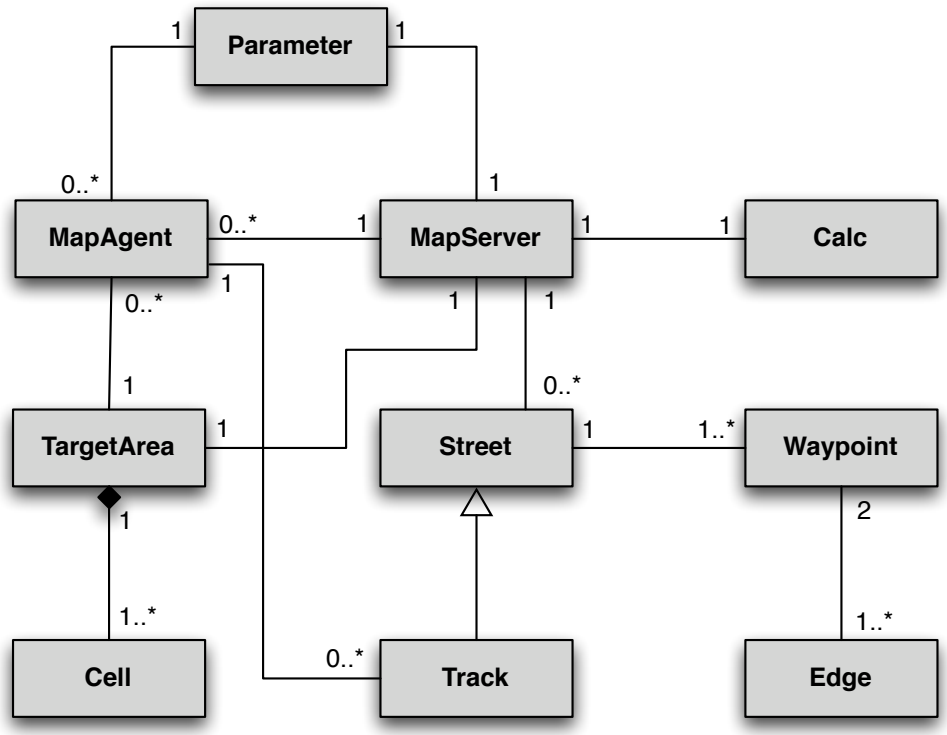


Abbildung 4.2: Klassendiagramm der Systemarchitektur

Um einen groben Eindruck darüber zu geben, wie die Konzepte durch die einzelnen Klassen umgesetzt wurden, ist an dieser Stelle eine kurze Erläuterung zu den Klassen aus Abbildung 4.2 zu finden. Dabei wurden auf Details verzichtet, da die genauen Beschreibungen der

Konzepte schon an den entsprechenden Stellen im Konzeptteil zu finden sind.

MapServer Ist die zentrale Klasse des Systems. Sie verwaltet alle bisher erfassten Straßen (Street-Objekte), aktualisiert den Zielbereich (TargetArea-Objekt) bei Erfassung einer neuen Straße und kommuniziert mit den Mobilgeräten (MapAgent-Objekte). Des Weiteren ist in dieser Klasse das Fusionsverfahren aus Abschnitt 3.2.2 implementiert, um mehrere GPS-Tracks miteinander zu vergleichen und entsprechend zu verarbeiten.

MapAgent Ein Objekt dieser Klasse stellt ein Mobilgerät dar, welches GPS-Ortungen durchführt und diese zu einem Track kombiniert. Diesen Track sendet es an das MapServer-Objekt und schickt bzw. empfängt gegebenenfalls Ad-hoc-Nachrichten.

Parameter Diese Klasse speichert alle systemrelevanten Parameter und stellt sie den anderen Klassen zur Verfügung. Die Belegung der Parameter kann dabei direkt über die OTcl-Skriptdatei geändert werden, so dass für die Simulation unterschiedlicher Szenarios keine erneute Kompilierung stattfinden muss. Um welche Parameter es sich genau handelt, wird im Rahmen der Evaluation ausführlich erläutert.

Calc Diese Hilfsklasse stellt der MapServer-Klasse mathematische Grundfunktionen bereit, welches statisch zugänglich sind.

Street Bildet die grundlegende Datenstruktur für eine Straße, welche vom MapServer-Objekt gespeichert wird. Sie besteht aus einer Reihe von Wegpunkten (Waypoint-Objekte), welche in einem Vektor geordnet gespeichert sind.

Track Bildet dahingehend eine Erweiterung der Street-Klasse, dass sie zusätzlich Informationen darüber beinhaltet, von welchem Mobilgerät der entsprechende Track erfasst wurde.

Waypoint Beschreibt einen GPS-Wegpunkt, indem er alle nötigen Informationen wie Koordinaten, Varianz der Messung und Geschwindigkeit enthält.

Edge Verbindet zwei Waypoint-Objekte und wird für die Erstellung der Histogramme bei der Fusion von Tracks benötigt.

TargetArea Der Zielbereich verwaltet alle darin befindlichen Zellen (Cell-Objekte), die über Methoden manipuliert werden können. Bei Eingang einer neuen Straße, werden alle darin befindlichen Zellen aktualisiert und der Status jeder einzelnen Zelle wird auf aktiv bzw. passiv gesetzt.

Cell Steht für eine Gitterzelle des Zielbereichs. Speichert alle mit der Zelle assoziierten Straßenabschnitte ab und kann mit diesen Informationen den eigenen Qualitätswert bestimmen, welcher das TargetArea-Objekt nutzt, um die Zelle zu aktivieren oder zu deaktivieren.

Neben dieser statischen Betrachtungsweise, werden wir im folgenden Abschnitt kurz auf einige Abläufe des Systems näher eingehen.

4.2.3 Ablaufdiagramme

Um die Zusammenhänge der zuvor vorgestellten Klassen besser verstehen zu können, werden an dieser Stelle die wichtigsten Systemabläufe in Form von Sequenz- bzw. Aktivitätsdiagrammen dargestellt und anschließend erläutert.

Verarbeitung eines neuen Tracks

Abbildung 4.3 zeigt die Sequenz von Objektinteraktionen, nachdem beim Server (MapServer) ein Track zur Verarbeitung von einem Mobilgerät (MapAgent) eingegangen ist. Der Server verarbeitet diesen Track und fügt ihn entweder zu einer bereits erfassten Straße hinzu oder nimmt ihn als neue Straße auf. Danach wird die neue bzw. veränderte Straße an das TargetArea-Objekt übergeben, welches alle Zellen ermittelt, durch welche diese Straße läuft. Für jede dieser Zelle ruft es eine Update-Methode auf, welche eine Zelle dazu veranlasst ihre Qualitätsmetrik zu aktualisieren und den neuen Wert dem TargetArea-Objekt zurückzugeben. Sind alle Qualitätswerte aktualisiert, weißt das Server-Objekt das TargetArea-Objekt an, für alle Zellen zu überprüfen, ob diese nach der Aktualisierung aller Metriken nun aktiv oder passiv sind und deren Status entsprechend zu ändern. Nach Abschluss dieses Vorgangs sendet der Server ein aktualisiertes TargetArea-Objekt zurück an alle Mobilgeräte, die sich in einer Zelle befinden, deren Status sich geändert hat bzw. die sich in einer Nachbarzelle einer solchen Zelle befinden.

Positionserfassung der Mobilgeräte

Abbildung 4.4 zeigt den Ablauf, den jedes Mobilgerät nach einer Positionsbestimmung durchläuft. Nachdem das Gerät seine Position ermittelt hat, stellt es fest in welcher Zelle sich diese Position befindet und prüft deren Status. Je nach Zell- und Gerätestatus ändert das Gerät seinen Status, startet bzw. beendet das Aussenden von Ad-hoc-Nachrichten und schickt gegebenenfalls den bis dato erfassten Track zum Server. Ist das Gerät aktiv fügt es die ermittelte Position zum aktuellen Track hinzu und wartet t_{GPS} Sekunden bis zur

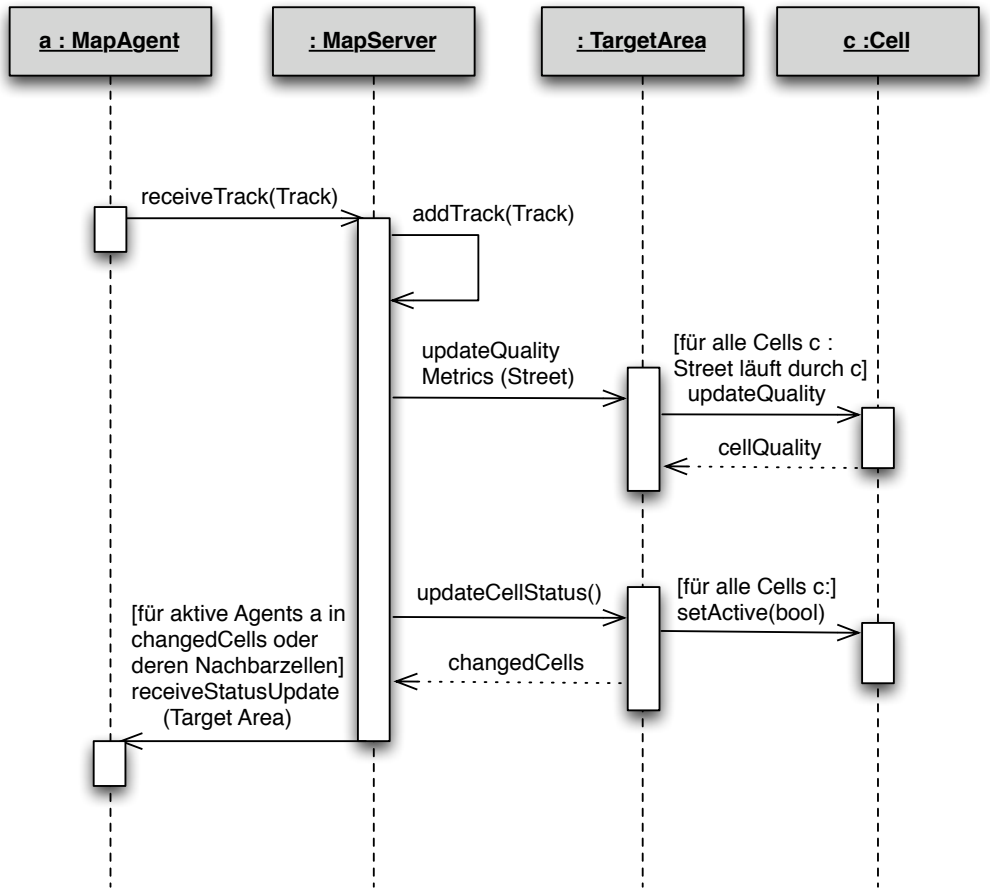


Abbildung 4.3: Sequenzdiagramm zur Verarbeitung eines neuen Tracks

nächsten Positionsüberprüfung. Ist das Gerät passiv, setzt es die Positionsbestimmung solange aus, bis es eine Ad-hoc-Nachricht erhält oder bis die Zeitspanne t_{thre} abgelaufen ist. Diese gibt die obere Schranke für die Wartezeit an, um den Fall abzufangen, dass in der gegenwärtige Zelle keine aktiven Mobilgeräte sind, die Ad-hoc-Nachrichten aussenden.

Nachdem die letzten zwei Abschnitte grob beschrieben haben, welche Konzepte in welchen Klassen umgesetzt wurden und wie einzelne Systemabläufe realisiert wurden, wird der folgende Abschnitt kurz darauf eingehen, welche Konzepte oder Ideen in der Implementierung keine Umsetzung fanden, da sie zur Evaluation der Konzepte dieser Arbeit nicht direkt beitragen.

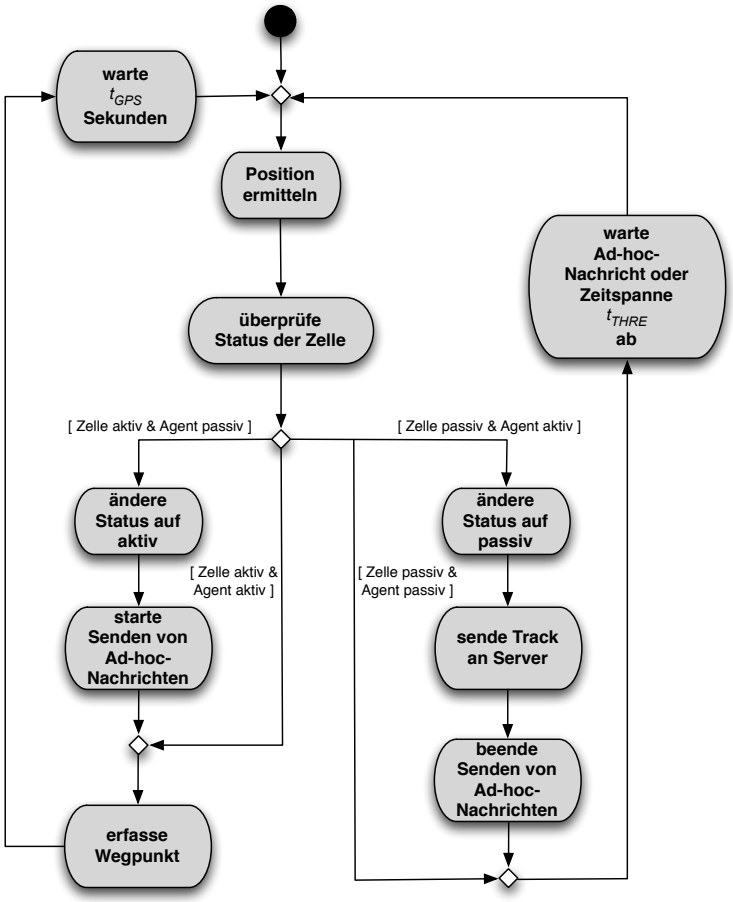


Abbildung 4.4: Positionserfassung der Mobilgeräte

4.2.4 Grenzen der Implementierung

Ziel der im nächsten Kapitel folgenden Evaluation ist es, anhand einer Simulation festzustellen, wie sich der Ressourcenverbrauch der Mobilgeräte im Verhältnis zu den Qualitätswerten der erfassten Straßen verhält. Da zum Nachweis dieses Ziels nicht alle in Kapitel 3 vorgestellten Konzepte benötigt werden, beschränkt sich die Implementierung auf die Bereiche, die für die Evaluation notwendig sind. Daher werden Konzepte, welche nicht implementiert wurden, an dieser Stelle kurz aufgeführt und eine kurze Begründung gegeben, warum auf ihre Umsetzung verzichtet werden konnte. Alle anderen Konzepte wurden in Anlehnung an die Beschreibungen aus dem Konzeptteil umgesetzt.

Lokationsmanagement Das in Abschnitt 3.4.2 eingeführte Lokationsmanagement wird nur benötigt, um bei Beginn der Erfassung festzustellen, welche Geräte sich in der Nähe des Zielgebiets aufhalten, um diesen den Auftrag zur Erfassung zu erteilen.

In der späteren Evaluation befinden sich jedoch alle Mobilgeräte, die simuliert werden, innerhalb der Karte des zu simulierenden Szenarios und somit in der Nähe des Zielbereichs. Daher ist die Menge der zu kontaktierenden Mobilgeräte von Anfang an bekannt und ein Lokationsmanagement, welches die Menge dieser Geräte anhand ihrer Position ermittelt, wird überflüssig.

Kategorisierung von Straßen Um eine Evaluation der in Abschnitt 3.4.9 definierten Kategorisierung von Straßen sinnvoll durchführen zu können, ist ein Vergleich zwischen einem Ist- und Sollzustand der kategorisierten Straßen nötig. In ns-2 sind Straßen nicht kategorisiert, so dass die Implementierung der Kategorisierung von Straßen zu keinem Erkenntnisgewinn bei der Evaluation beitragen würde.

Partizipation des Nutzers Die in den Optimierungspotentialen in Abschnitt 3.4.10 beschriebene Partizipation des Benutzers, setzt den interaktiven Einsatz von Teilnehmern voraus, um die von ihnen erfassten GPS-Daten anzureichern oder gezielt bisher nicht erfasste Bereiche zu erschließen. Dieser Ansatz lässt sich in ns-2 nur sehr schwer umsetzen, da sich die Mobilgeräte anhand eines Tracefiles auf fest definierten Routen bewegen und auch keine genaueren Informationen über die Straßen besitzen auf denen sie sich bewegen. Da das Ziel dieser Arbeit sowieso in der Umsetzung und Evaluation eines opportunistischen Ansatzes liegt, wird auf die Implementierung von partizipatorischen Elementen verzichtet.

Vereinfachung von Trajektorien Das ebenfalls als optional gekennzeichnete Konzept zur Vereinfachung von Trajektorien auf Seiten der Mobilgeräte wurde nicht implementiert. Wie bereits im zugehörigen Abschnitt 3.4.10 erwähnt, müsste zuerst ein Verfahren entwickelt werden, welches möglichst ohne Informationsverlust bei wichtigen Informationen wie der Varianz der Wegpunkte arbeitet, da ansonsten die zuvor definierten Qualitätsmetriken nicht dem genauen Bild der Erfassung entsprechen würden.

4.3 Zusammenfassung

Das vorliegende Kapitel der Implementierung gab eine kurze Einführung zum Netzwerksimulator ns-2, sowie zur Umsetzung der im Konzeptteil vorgestellten Methoden.

Das nun folgende Kapitel der Evaluation soll die Frage untersuchen, in wie weit die entwickelten Konzepte wirklich zu einer Einsparung von Ressourcen führen können und wie sie sich auf die resultierende Qualität der erstellten Karte auswirken.

Kapitel 5

Evaluation

Um die in dieser Masterarbeit vorgestellten Konzepte dahingehend zu prüfen, ob sie die gewünschten Resultate erzielen, beschreibt diese abschließende Evaluation die Ergebnisse von Simulationen, die auf Basis des in der Implementierung umgesetzten Systems stattfanden. Das Ziel besteht in erster Linie darin, zu überprüfen, wie sich die entwickelten Konzepte auf den Ressourcenverbrauch und die Qualität der zu erstellenden Straßenkarte auswirken. Um dies beurteilen zu können, vergleichen wir die Ergebnisse der durchgeführten Simulationen mit den Ergebnissen einer unkoordinierten Erfassung, bei der alle Mobilgeräte dauerhaft GPS-Daten erfassen und keine zentrale Steuerung der Erfassung vorliegt.

Beginnend mit einer genaueren Erläuterung, wie realistische Bewegungsabläufe von Fußgängern und Autos in ns-2 umgesetzt werden können und welche Szenarios wir für die später folgende Simulation verwenden, leitet Abschnitt 5.1 dieses Kapitel ein. Im darauf folgenden Abschnitt 5.2 werden die Ein- und Ausgabeparameter definiert, welche festlegen, mit welchen Parametern die Simulationen gestartet werden bzw. anhand welcher Werte die Ergebnisse der Simulationen beurteilt werden sollen. Danach beginnen in Abschnitt 5.3 die eigentlichen Simulationen. Hier werden, für die zuvor definierten Szenarios und Eingabeparameter, die aus den Simulationen resultierenden Ausgabewerte analysiert und ausgewertet. Die daraus gewonnenen Ergebnisse werden abschließend in Abschnitt 5.4 noch mal zusammengefasst und der Versuch unternommen, eine allgemeine Aussage aus diesen abzuleiten.

5.1 Tracefiles

In diesem Abschnitt wollen wir die für eine Simulation nötigen Tracefiles näher beschreiben und die Parameter festlegen, welche zur Erstellung dieser notwendig sind. Dazu gibt

Abschnitt 5.1.1 eine Einführung in die wichtigsten Eigenschaften dieser Files, um die im danach folgenden Abschnitt 5.1.2 konkreten Parameterbelegungen, welche die Szenarios der Simulationen festlegen, verstehen zu können.

5.1.1 Grundlagen

Ns-2 unterstützt bei der Simulation die Einbindung von so genannten Tracefiles. Ein solches Tracefile kann jedem ns-2-Objekt zu einem bestimmten Zeitpunkt eine Position, über die Angabe von X- und Y-Koordinaten, zuweisen. Diese Koordinaten können von dem jeweiligen Objekt ausgelesen werden, so dass dadurch die Bewegungsabläufe von Mobilgeräten simuliert werden können. Die Tracefiles, mit den darin beschriebenen Bewegungsabläufen, werden von einem Tracefile-Generator erstellt, welcher anhand einer internen Straßenkarte diese Bewegungsabläufe generiert. So entsteht eine realistische Simulationsumgebung für das Systems. Da die vom Tracefile generierten Koordinaten jedoch immer exakt auf einer im Generator hinterlegten Straße liegen, wurde in der Implementierung ein zufälliger GPS-Fehler zu diesen Koordinaten hinzugefügt, um so eine realistische Simulation einer GPS-Ortung zu ermöglichen.

Ein Generator für solche Tracefiles ist *UDel Models* [28], welcher annähernd realistische Szenarios für mobile Netzwerke innerhalb von Städten generiert. Dieser Generator wird im Rahmen dieser Evaluation für die Erstellung der nötigen Tracefiles genutzt, mit deren Hilfe die Bewegungen von Fußgänger und Autos innerhalb einer Stadt so realistisch wie möglich nachgebildet werden sollen. UDel Models benötigt für die Generierung der Tracefiles eine Reihe von Parametern, welche die Eigenschaften des zu erstellenden Szenarios beeinflussen. Da es eine Vielzahl dieser Parameter gibt, werden an dieser Stelle nur diejenigen erwähnt, die im Rahmen dieser Evaluation von Bedeutung sind.

- Mobile Objekte werden in UDel Models durch Fußgänger oder Autos simuliert. Für beide ist festzulegen, wie viele davon im Szenario vorkommen und wie schnell sich diese maximal bewegen sollen.
- Der Zeitraum und die Länge des Zeitraums, über den sich das Szenario erstreckt, kann durch die Angabe einer Start- und Enduhrzeit festgelegt werden.
- Da die Daten des erstellten Tracefiles einen zufälligen Charakter haben sollen und jeweils verschiedene Bewegungsabläufe enthalten sollen, nutzt das Programm einen Random-Seed als Initialwert für die probabilistischen Funktion, welche das Tracefile erstellt.

Mit welchen Werten wir diese Parameter bei der Generierung der Tracefiles belegen werden, zeigt der nachfolgende Abschnitt.

5.1.2 Erstellung von Tracefiles

Grundlage der Simulationen sind die Tracefiles, welche die Bewegungsmuster angeben, mit welchen sich die mobilen Geräte durch eine virtuelle Stadt bewegen. Dafür erstellen wir zehn Tracefiles, welche die in Tabelle 5.1 aufgeführten Belegungen als Eingangsgrößen für den Tracefile-Generator haben.

Anzahl	#Fußgänger	#Autos	Zeitraum	Seed
5	1000	0	05:00 - 23:30	$\{1278693248, \cdots, 1278693252\}$
5	0	1000	05:00 - 23:30	$\{1279204368, \cdots, 1279204372\}$

Tabelle 5.1: Parameterbelegung für Simulation

Wie aus der Tabelle zu sehen ist, handelt es sich um insgesamt zehn unterschiedliche Tracefiles, die sich in zwei Gruppen zu jeweils fünf Tracefiles untergliedern und sich innerhalb dieser Gruppe jeweils nur in der Belegung des Seeds unterscheiden. Die erste Gruppe beschreibt ein Szenario, welches die Bewegungen von Fußgängern simuliert, die zweite Gruppe die von Autos. Jede dieser Gruppen enthält fünf unterschiedliche Werte für den Seed, so dass mehrere Bewegungsmuster für die ansonsten gleiche Konfiguration entstehen und somit vermieden wird, dass Effekte, welche nur für ein spezielles Tracefile auftreten, in der Simulation als Ergebnis gedeutet werden. Für die Erstellung der Tracefiles wird die in UDel Models beinhaltete Straßenkarte von Chicago verwendet (*Chicago2kmx2km*).

Jeder der in diesem Abschnitt beschriebenen Simulationen wird für alle zehn Tracefiles einzeln durchgeführt. Dadurch erhalten wir für jede Simulation zehn verschiedene Ergebnisse. Um mit diesen Arbeiten zu können, bilden wir aus den fünf Szenarios für Fußgänger bzw. Autos den Mittelwert und erhalten so Ergebnisse, die aus mehreren Messreihen hervorgehen und jeweils die Eigenheiten der Simulation für Fußgänger bzw. für Autos beinhalten.

Um eine realistische Simulation zu ermöglichen, müssen wir außerdem eine weitere Eigenschaft der Tracefiles berücksichtigen, welche die Bewegungsmuster der mobilen Knoten betrifft. Diese bewegen sich nicht nur auf den Straße der in UDel Models hinterlegten Karte, sondern können sich auch innerhalb von Gebäuden befinden, so dass sich ihre Position für längere Zeit nicht ändert. Da wir die dabei entstehenden GPS-Ortungen beim Ergebnis der Simulation jedoch nicht einkalkulieren wollen, weil dieser Fall in der Realität normalerweise durch das fehlende GPS-Signal erkannt wird, wollen wir für die folgenden Simulationen keine GPS-Ortung berücksichtigen, die durchgeführt werden, wenn der mobile Knoten auf der gleichen Stelle stehen bleibt. Das führt wiederum dazu, dass auch GPS-Ortungen eines stillstehenden Geräts nicht berücksichtigt werden, wenn sich das Gerät außerhalb eines Gebäudes befindet. Diese Eigenschaften würden dem Einsatz des, in Abschnitt 3.4.10 als Optimierung vorgeschlagenen, Beschleunigungssensors gleichkommen.

5.2 Parameter

Ausschlaggebend für das Ergebnis der Simulationen sind die Eingabeparameter, welche das Verhalten des Systems steuern. Um herauszufinden, wie sich das Systems für verschiedene Belegungen der Eingabeparameter verhält, muss idealerweise eine Vielzahl an möglichen Kombinationen dieser Belegungen simuliert werden. Bevor dies geschehen kann, müssen diese Parameter jedoch eindeutig definiert werden.

Auf der anderen Seite steht das Ergebnis der Simulation. Dieses drückt sich in Form von Ausgabewerten aus, die das Verhalten des Systems, hinsichtlich der zu überprüfenden Eigenschaften, beschreiben. Um die Menge dieser beiden Metriken einheitlich zu beschreiben, bietet der folgende Abschnitt eine Übersicht über die für die Evaluation genutzten Eingabeparameter, sowie den Ausgabewerten, die das System am Ende einer Simulation liefert.

5.2.1 Eingabeparameter

Wie bereits im Konzeptteil deutlich wurde, enthält das implementierte System eine Reihe von Parametern, welche das Verhalten des Systems entscheidend beeinflussen. Diese werden nachfolgend in Tabelle 5.2 zusammengefasst, eine jeweilige Referenz auf die entsprechenden Abschnitte im Konzeptteil gegeben und anschließend noch einmal kurz erläutert.

Symbol	Parameter	Abschnitt
l_g	Seitenlänge einer Gitterzelle	3.2.1
b_r	Breite des Randbereichs für Ad-hoc-Kommuinkation	3.4.6
q_{rel}	Delta für Bestimmung des Zellstatus	3.4.7
I_{GPS}	Intervall zwischen GPS-Ortungen	3.1.3
I_{adhoc}	Intervall zwischen Senden von Ad-hoc-Nachrichten	3.4.6

Tabelle 5.2: Eingabeparameter für Simulation

l_g Legt die Seitenlänge der Gitterzellen fest, die den Zielbereich unterteilen.

b_r Legt die Breite des Randbereichs innerhalb einer Gitterzelle fest, innerhalb welcher die Mobilgeräte Ad-hoc-Nachrichten verschicken, um passiven Mobilgeräten das Betreten einer aktiven Zelle mitzuteilen.

q_{rel} Legt den Wert des Deltas fest, welches der Server nutzt um Entscheidungen darüber zu treffen, ob eine Zelle aktiven oder passiven Status erhält.

I_{GPS} Legt den zeitlichen Abstand fest, der zwischen zwei aufeinander folgenden GPS-Ortungen eines Mobilgeräts liegt, wenn dieses aktiv ist und daher gerade einen Track erfasst.

I_{adhoc} Legt den zeitlichen Abstand fest, der zwischen dem Senden von zwei aufeinanderfolgenden Ad-hoc-Nachrichten liegt, wenn sich das Mobilgerät im Randbereich einer Zelle befindet und aktiv ist.

Diese Vielfalt an Parametern und deren breites Spektrum an möglichen Wertebereichen, ergeben eine sehr große Möglichkeit an Kombinationen für ihre Belegungen. Daher werden wir in Abschnitt 5.4 zuerst ein plausibel erscheinendes Standardszenario festlegen, auf dessen Basis wir einzelne Belegungen der Parameter untersuchen können.

5.2.2 Ausgabewerte des Systems

Wie bereits an mehreren Stellen erwähnt, besteht das Ziel dieser Arbeit darin, durch das entworfene Konzept zu zeigen, dass mit der darin vorgestellten Technik eine Erfassung von Straßenkarten möglich ist, bei der möglichst wenig Ressourcen der Mobilgeräte verbraucht werden, aber dennoch die erzielte Qualität der erfassten Straßen darunter nicht all zu stark leidet. Um diese Aussagen nachprüfen zu können, brauchen wir zum einen eine Metrik, welche die Qualität der erstellten Straßenkarte angibt, und zum anderen Metriken für den Verbrauch der Ressourcen der Mobilgeräte.

Ausgabewerte für Qualitätsaussagen

Das Ziel der nachfolgenden Simulationen besteht darin, einen Vergleich zwischen den Ergebnissen der Konzepte dieser Arbeit — der gezielten Erfassung — und der trivialen Erfassung herzustellen. Im trivialen Fall findet keine Koordination der Erfassung statt, so dass alle Mobilgeräte dauerhaft GPS-Ortungen durchführen und diese an den Server schicken. Um nun eine Aussage über die erstellte Kartenqualität abzuleiten, vergleichen wir die Anzahl der Wegpunkte, die am Ende der Simulation bei der trivialen bzw. der gezielten Erfassung die resultierende Karte beschreiben, da jeder Wegpunkt eine zusätzliche Information für die erstellte Karte bedeutet. Dabei ist zu beachten, dass nicht jeder GPS-Wegpunkt eines Tracks in einem Wegpunkt in der erstellten Karte resultiert. Beschreibt der erfasste GPS-Track zum Beispiel Wegpunkte auf einer Straße, die bereits im Straßenmodell vorhanden sind, so werden diese mit bereits vorhandenen Punkten aus dem Straßenmodell fusioniert und bilden daher keine zusätzlichen Wegpunkte im finalen Straßenmodell.

Neben dieser quantitativen Betrachtung wollen wir zusätzlich einen Blick auf die Qualität der erfassten Straßen werfen, indem wir die in Abschnitt 3.3.1 vorgestellte Qualitätsmetrik

nutzen, um für die Straßen, welche sich am Ende der Erfassung im Straßenmodell befinden, einen Qualitätswert zu ermitteln.

Ausgabewerte für den Ressourcenverbrauch

Der Ressourcenverbrauch der Mobilgeräte kategorisiert sich in die zwei Gruppen GPS-Ortungen und Kommunikationskosten, da diese beiden die vom System beeinflussten Größen auf einem Mobilgerät sind. Während die Anzahl der GPS-Ortungen eindeutig ist, können die Kommunikationskosten weiter unterteilt werden. Wir unterscheiden hier zwischen den Kosten, die durch das Verschicken von Ad-hoc-Nachrichten entstehen, und den Kosten, die durch die Kommunikation zwischen Server und Mobilgeräten entstehen. Dadurch ergeben sich die in Tabelle 5.3 beschriebenen Operationen. Um diesen eine konkrete Gewichtung zu geben, ist zu jeder ein realistischer Ressourcenverbrauch angegeben [47], wobei sich der Ressourcenverbrauch der Nachrichten jeweils auf eine Größe von 1000 Bit bezieht.

Symbol	Operation	Energieverbrauch in mJ
op_{GPS}	GPS-Ortung	75
$op_{A_{out}}$	Ad-hoc-Nachricht versenden	2
$op_{A_{in}}$	Ad-hoc-Nachricht empfangen	1
$op_{S_{out}}$	Nachricht an Server senden	80
$op_{S_{in}}$	Nachricht von Server empfangen	40

Tabelle 5.3: Energieverbrauch eines Mobilgeräts

Bevor wir aus diesen Operationen die Metriken für den Ressourcenverbrauch ableiten können, muss für jede überlegt werden, wie sich diese auf den Ressourcenverbrauch insgesamt auswirkt. Dazu nehmen wir die bereits vorgestellte triviale Erfassung zum Vergleich. Dies führt zu folgenden Feststellungen:

- Ad-hoc-Nachrichten werden bei der trivialen Erfassung nicht versendet. Ihr Ressourcenverbrauch wird daher nicht in den Gesamtenergieverbrauch der trivialen Erfassung eingerechnet.
- In beiden Szenarios werden Nachrichten an den Server geschickt, jedoch in unterschiedlichem Ausmaß. Da beim trivialen Fall mehr GPS-Wegpunkte aufgezeichnet werden, ist die Menge der zu übertragenden Daten insgesamt größer. Im Gegenzug dazu ist bei der gezielten Erfassung die Anzahl der gesendeten Nachrichten höher, so dass der Overhead für das Versenden größer ist. Da der Zeitabstand, in welchem

die Mobilgeräte ihre aufgezeichneten GPS-Tracks an den Server schicken, im trivialen Fall nicht festgelegt ist und somit ein direkter Vergleich des Energieverbrauchs schwierig ist, verwenden wir folgende Vereinfachung: Wir gehen davon aus, dass sich die erhöhte Menge der zu übertragenden Daten im trivialen Fall durch den Mehraufwand an Nachrichten-Overhead für den Fall der gezielten Erfassung ausgleicht und vernachlässigen die Anzahl der an den Server geschickten Nachrichten bei der Betrachtung der verbrauchten Energie in beiden Fällen.

- Nachrichten, die vom Server zurück an die Mobilgeräte geschickt werden, fallen bei der trivialen Erfassung nicht an, da diese bei der erzielten Erfassung dafür genutzt werden, um den Mobilgeräten Statusänderungen der Gitterzellen mitzuteilen. Ihr Ressourcenverbrauch wird nicht in den Gesamtenergieverbrauch der trivialen Erfassung eingerechnet.

Auf Basis dieser Überlegungen definieren wir für die triviale (Gleichung 5.1) bzw. koordinierte (Gleichung 5.2) Erfassung den Gesamtverbrauch an Energie in mJ, der sich aus den Kosten der Operationen aus Tabelle 5.3 und der Anzahl der durchgeführten Operationen ergibt. Dabei ist beim Gesamtverbrauch des trivialen Falls nur die Anzahl der GPS-Ortungen ausschlaggebend, da die anderen Operationen, wie in der oben stehenden Aufzählung erklärt wurde, im trivialen Fall nicht benötigt werden.

$$EC_t = \#op_{GPS} \cdot 75 \tag{5.1}$$

$$EC_c = \#op_{GPS} \cdot 75 + \#op_{A_{in}} \cdot 1 + \#op_{A_{out}} \cdot 2 + \#op_{S_{in}} \cdot 40 \tag{5.2}$$

Mit Hilfe dieser beiden Größen, können wir nun für die jeweiligen Simulationen direkt den Energieverbrauch ermitteln.

5.3 Simulation

Nachdem die Tracefiles erstellt und die Ein- bzw. Ausgabeparameter definiert wurden, kann mit der eigentlichen Simulation des Systems begonnen werden. Wie aus Abschnitt 5.2.1 bereits hervorging, ermöglicht die Vielzahl an Systemparametern eine große Anzahl an möglichen Szenarios die zu simulieren sind. Da in diesem Abschnitt nicht auf jedes Ergebnis einzeln eingegangen werden kann, werden wir die Erkenntnisse aus den Simulationen in Relation zu einem Standardszenario erklären, für das wir plausible Belegungen für die Eingabeparameter festlegen. Dieses wird in Abschnitt 5.3.1 definiert. Danach beginnen

wir in Abschnitt 5.3.2 mit den eigentlichen Simulationen, deren wichtigsten Ergebnisse vorgestellt werden. Zu guter Letzt werden die Ergebnisse aller Simulationen in Abschnitt 5.4 analysiert und versucht allgemeine Aussagen zu treffen, die sich aus diesen ableiten lassen.

5.3.1 Standardszenario

Wir beginnen mit der schon erwähnten Festlegung eines Standardszenarios, indem wir für die Parameter aus Tabelle 5.2 Werte festlegen, die einem realistischen Szenario entsprechen könnten und sich in den durchgeführten Simulationen als plausible Belegungen erwiesen. Daher wird in Tabelle 5.4 für jeden Parameter einen Standardwert festgelegt.

Symbol	Parameter	Standardwert
l_g	Seitenlänge Gitterzelle	200m
b_r	Breite Randbereich	20m
q_{rel}	Delta für Statusentscheidung	0.01
I_{GPS}	Intervall GPS-Ortung	10s
I_{adhoc}	Intervall zwischen Senden von Ad-hoc-Nachrichten	300s

Tabelle 5.4: Parameterbelegung Standardszenario

Durch die Definition eines Standardszenarios ist es uns nun möglich, verschiedene Werte für einzelne Parameter zu simulieren. Durch die Verwendungen mehrere Tracefiles pro Parameterbelegung, kann vermieden werden, dass einzelne Ergebnisse, durch die Eigenheiten eines Tracefiles, zufällig entstehen. Dabei wurde für alle Eingabeparameter eine möglichst große Auswahl an Belegungen simuliert. Jede Parameterbelegung wurde jeweils für die zwei in Abschnitt 5.1.2 definierten Szenarios für Fußgänger bzw. Autos simuliert. Die wichtigsten Erkenntnisse aus diesen Ergebnissen werden im nächsten Abschnitt vorgestellt.

5.3.2 Simulation

Im Rahmen der Simulation wurden alle vorgestellten Parameter mit mehreren Werten belegt und die daraus entstandenen Kombinationen simuliert. Da nicht jeder der Parameter einen signifikanten Effekt auf die zuvor definierten Ausgangsgrößen hatte, werden nachfolgend nur die wichtigsten Ergebnisse vorgestellt. Dabei wird in erster Linie untersucht, ob das vorgestellte Konzept das gewünschte Verhalten für die einzelnen Ausgangsgrößen liefert und in wieweit es möglich ist, Ressourcen bei der Erfassung einzusparen.

Die größte Auswirkung auf das Ergebnis, über alle Simulationen hinweg, hat die Wahl des Parameters q_{rel}. Dieser legt fest, wie der Server entscheidet, welche Zellen passiv oder aktiv

zu setzen sind. Für die folgenden Simulationen wurden daher die Parameterbelegung aus dem Standardszenario gewählt und q_{rel} über die folgenden Werte variiert:

$$q_{rel} \in \{0.01, 0.05, 0.1, 0.3, 0.5, 1\}$$

Hierbei bildet der Fall $q_{rel} = 1$ das bereits erwähnte triviale Szenario ab, bei dem dauerhaft GPS-Ortungen von den Mobilgeräten durchgeführt werden.

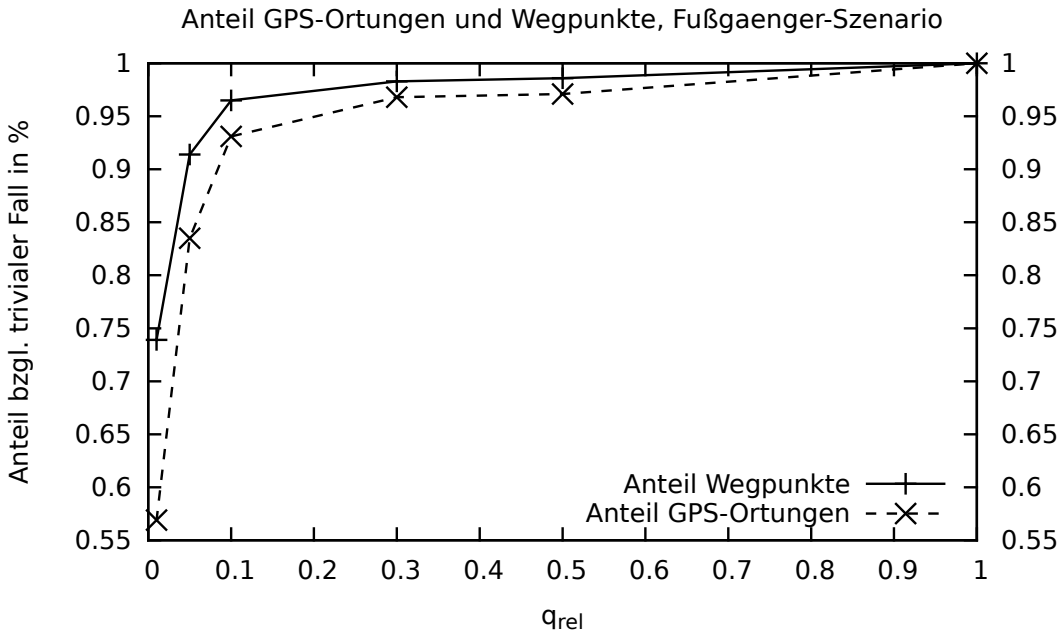
Wir unterteilen die Simulationen in mehrere Abschnitte. Zuerst wollen wir uns die Ergebnisse der Simulationen hinsichtlich der erzielten Qualität der Straßenkarte anschauen. Danach untersuchen wir den Ressourcenverbrauch der Mobilgeräte und Vergleichen im Anschluss diese Ergebnisse miteinander.

Analyse der Qualität

Zuerst wollen wir uns in Abbildung 5.1, für unterschiedliche Werte von q_{rel}, den Anteil der durchgeführten GPS-Ortungen und den Anteil der erfassten Wegpunkte, bezüglich der trivialen Erfassung (= 100%), für das Fußgänger- und Auto-Szenario anschauen. Zuerst ist zu beobachten, dass der Anteil der GPS-Ortungen, sowie der Anteil der erfassten Wegpunkte, im Vergleich zum trivialen Fall für kleine q_{rel} gering ist und dann mit größer werdendem q_{rel} steigt. Die Begründung hierfür ist leicht nachzuvollziehen. Da bei hohen Werten von q_{rel} weniger Zellen einen passiv Status erhalten, werden diese nicht von der Erfassung ausgesetzt und die Anzahl der GPS-Ortungen erhöht sich. Durch die höhere Anzahl an GPS-Ortungen werden auch mehr Wegpunkte von den Mobilgeräten erfasst. Diese fliesen jedoch nicht zwangsläufig als neue Wegpunkte ins Straßenmodell des Servers ein, da sie vom Server fusioniert werden können, wenn sie eine bereits erfasste Straße beschreiben. Dadurch erklärt sich die Differenz zwischen dem Anteil der GPS-Ortungen und dem der Wegpunkte in den Schaubildern.

Da die Kurve, die den Anteil der GPS-Ortungen beschreibt, in beiden Szenarios höher ist, als die Kurve, die für den Anteil der Wegpunkte steht, müssen Wegpunkte fusioniert worden sein, da die Kurven ansonsten identisch wären. Wie viele Fusionen in den Simulationen jeweils durchgeführt wurden, wollen wir uns im Folgenden anschauen.

Das erste Schaubild in Abbildung 5.2 zeigt die Anzahl der durchgeführten Fusionen für verschiedene Werte von q_{rel}. Wie zu sehen ist, steigt die Anzahl dieser Operation mit größer werdendem q_{rel} an. Beim Auto-Szenario steigt die Anzahl der Fusionen für große q_{rel} sogar sehr stark. Dies ist ein Anzeichen dafür, dass viele Wegpunkte, die von den Mobilgeräten erfasst wurden, auf dem Server nur noch fusioniert werden, da die zugehörige Straße bereits erfasst wurde. Der Anstieg bei der Zahl der Fusionen zeigt, dass viele der zusätzlichen GPS-Ortungen, die bei hohen Werten von q_{rel} durchgeführt werden (siehe



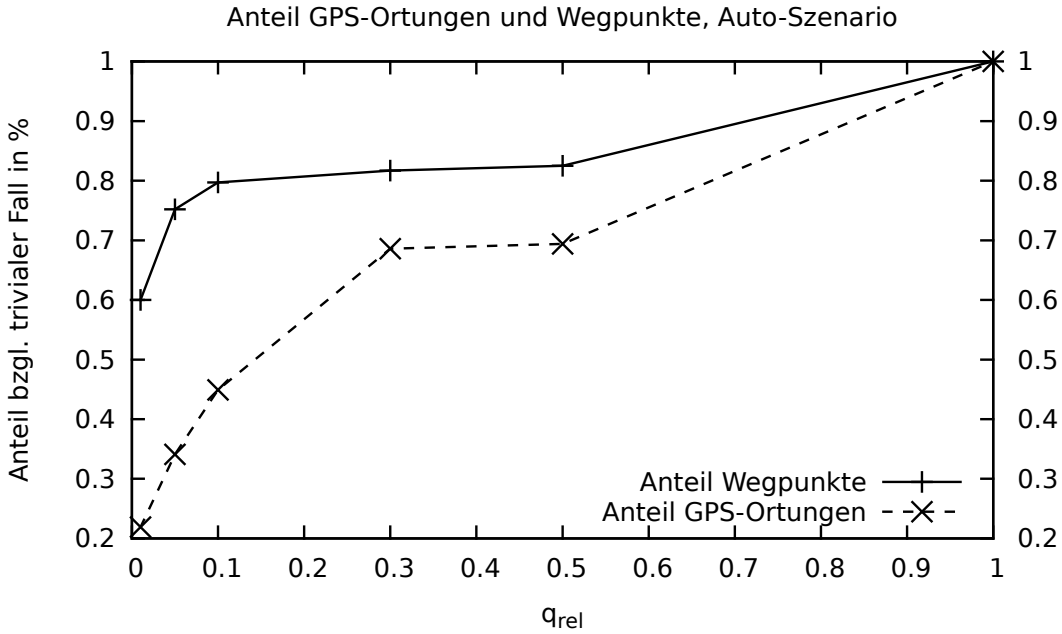


Abbildung 5.1: Anteil der Wegpunkte und #GPS-Ortungen

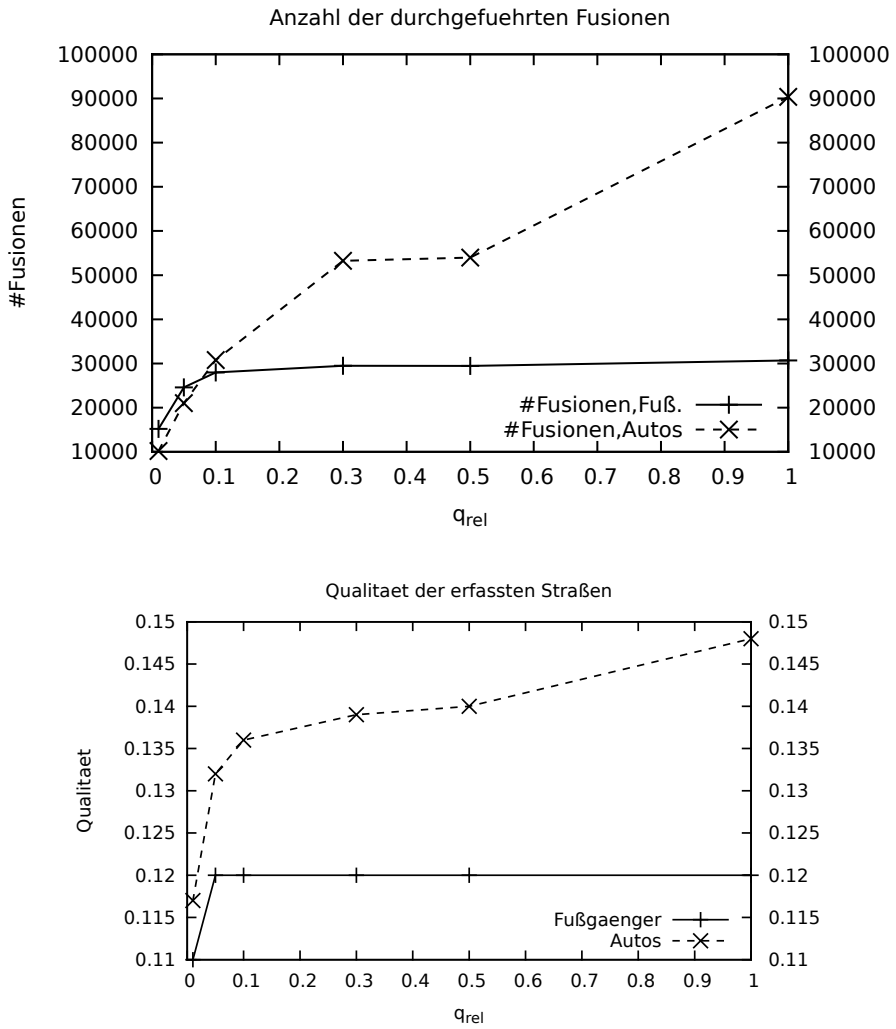


Abbildung 5.2: Anzahl der durchgeführten Fusionen und Straßenqualität im Vergleich

Abbildung 5.1), auf Straßen durchgeführt wurden, die bereits im Straßenmodell vorhanden waren. Nachfolgend wollen wir uns anschauen, welche Auswirkungen dies für die Qualität dieser bereits erfassten Straßen hat.

Als Indikator für die Qualität der erfassten Straßen nehmen wir die in Abschnitt 3.3.1 vorgestellte Qualitätsmetrik für Straßen. Das zweite Schaubild in Abbildung 5.2 zeigt den durchschnittlichen Qualitätswert der erfassten Straßen am Ende der Simulation. Dabei wird aus dem Schaubild ersichtlich, dass dieser im Auto-Szenario mit steigendem q_{rel} steigt und im Fußgänger-Szenario bis zu einem Wert von 0.1 steigt. Wie im Konzeptteil gezeigt wurde, steigt der Qualitätswert einer Straße mit jeder Fusion, die durchgeführt wird. Da wir im Schaubild davor gesehen haben, dass die Anzahl der Fusionen mit steigendem q_{rel}

ebenfalls steigt, ist hier der Zusammenhang erkennbar. Speziell im Auto-Szenario ist gut zu erkennen, dass durch die hohe Anzahl an Fusionen der durchschnittliche Qualitätswert der erfassten Straßen stärker zunimmt.

Analyse des Ressourcenverbrauchs

Die folgenden Betrachtungen sollen den Ressourcenverbrauch der Mobilgeräte näher untersuchen. Wie Abbildung 5.1 schon zeigte, steigt die Anzahl der GPS-Ortungen mit steigendem q_{rel} an. Nachfolgend wollen wir uns das Verhalten für die beiden anderen entscheidenden Größen anschauen — der Ressourcenverbrauch für die Ad-hoc-Nachrichten und für die Kommunikation mit dem Server.

Zuerst schauen wir uns an, wie viele Ressourcen für das Senden und Empfangen von Ad-hoc-Nachrichten verbraucht wurden. Das erste Schaubild in Abbildung 5.3 zeigt den Gesamtenergieverbrauch aller Mobilgeräte, der sich durch Addition des Energieverbrauchs für gesendete und empfangene Nachrichten der einzelnen Geräte ergibt. Wie zu sehen ist, nimmt in beiden Szenarios der Energieverbrauch dieser Nachrichten mit steigendem q_{rel} zu. Mobilgeräte versenden Ad-hoc-Nachrichten, wenn sie sich im Randbereich einer aktiven Zelle befinden, da mit steigendem q_{rel} die Anzahl der aktiven Zellen zunimmt, steigt auch der Energieverbrauch entsprechend. Im Vergleich dazu, werden bei der trivialen Erfassung keine Ad-hoc-Nachrichten versendet und es wird daher keine Energie verbraucht.

Das zweite Schaubild in Abbildung 5.3 zeigt die Anzahl der Nachrichten, die der Server an die Mobilgeräte sendet, um ihnen die Aktualisierung des Statusvektors zukommen zu lassen. Die Anzahl dieser Nachrichten nimmt mit steigendem q_{rel} ab, da weniger Zellen einen passiven Status erhalten und der Server eine Aktualisierungsnachricht nur schickt, wenn sich der Status mindestens einer Zelle geändert hat. Die Anzahl der gesendeten Nachrichten ist im trivialen Fall bei null.

Unter Hinzunahme der Anzahl der GPS-Ortungen aus Abbildung 5.1, können wir nun die Summe des Gesamtenergieverbrauchs aller Mobilgeräte ermitteln. Das Ergebnis ist im unteren Schaubild aus Abbildung 5.3 zu sehen. Wie vermutet, nimmt der Energieverbrauch mit steigendem q_{rel} zu. Dies begründet sich hauptsächlich durch die zunehmende Anzahl der GPS-Ortungen. Dabei wird für niedrige q_{rel} der Mehraufwand an Energie, der durch die Ad-hoc- und Servernachrichten entsteht, kompensiert, so dass insgesamt eine Ersparnis gegenüber dem trivialen Fall entsteht.

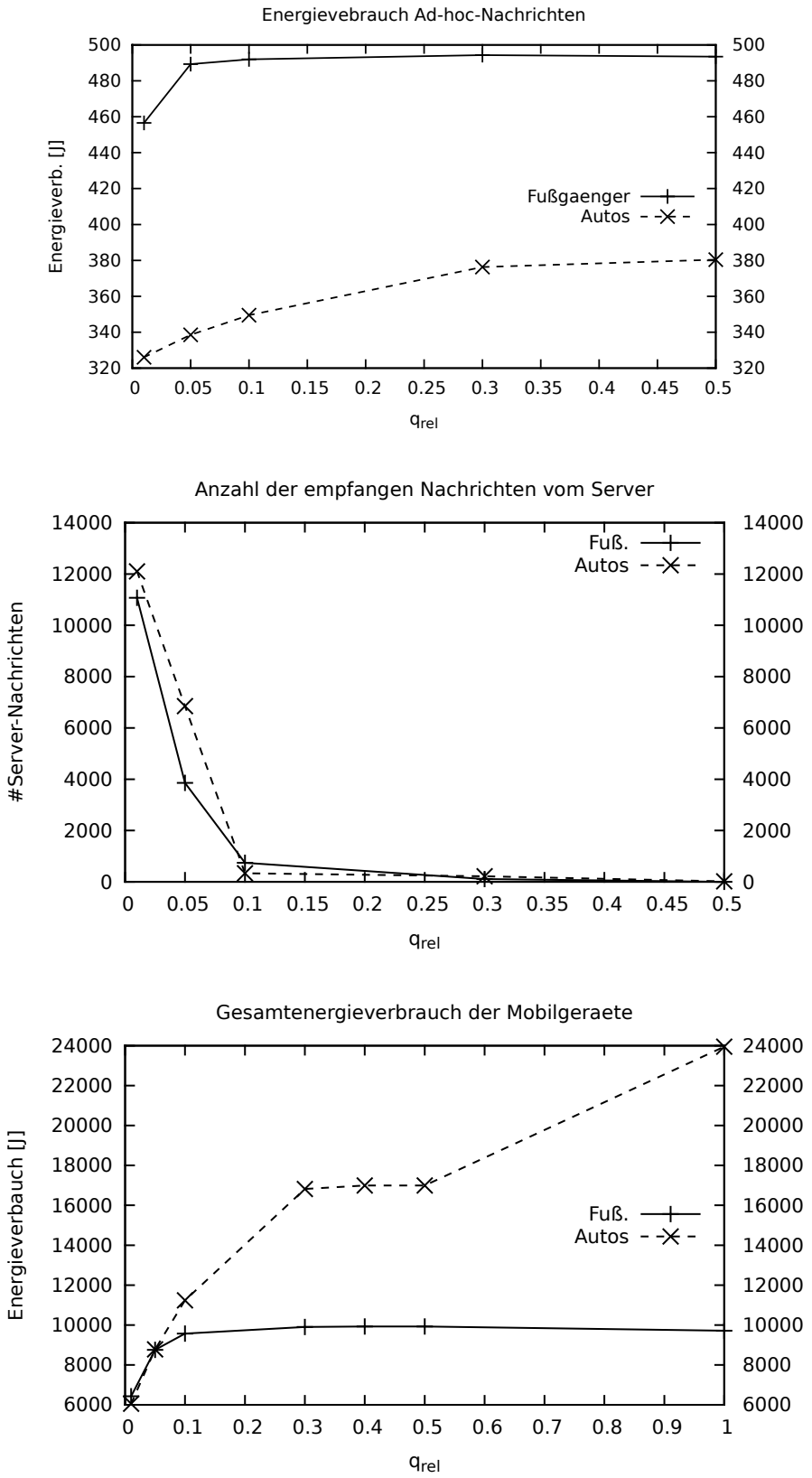


Abbildung 5.3: Energieverbrauch der Mobilgeräte

Vergleich der Ergebnisse

Nachdem wir die Ergebnisse der Simulationen bezüglich der Qualität und dem Ressourcenverbrauch diskutiert haben, wollen wir die erzielten Werte miteinander vergleichen. Mit Hinblick auf die triviale Erfassung, schauen wir uns die relative Energieeinsparung an und stellen diese dem relativen Qualitätsverlust gegenüber. Dabei bewerten wir die Qualität anhand des, in Abbildung 5.1 bereits betrachteten, relativen Anteils der erfassten Wegpunkte, anstelle der Qualitätsmetrik für erfasste Straßen, da wir unser Hauptaugenmerk auf die Vollständigkeit der erfassten Karte legen wollen. Die relative Energieeinsparung berechnen wir in Anlehnung an den Gesamtenergieverbrauch aus Abbildung 5.3.

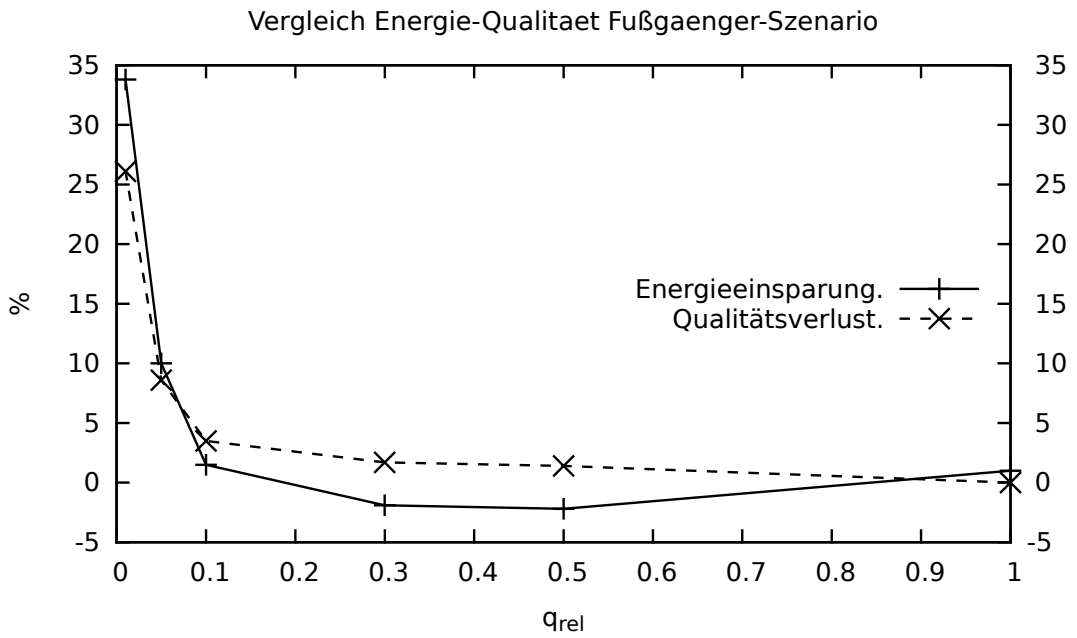
Abbildung 5.4 zeigt das Ergebnis dieser Gegenüberstellung. Wie zu erkennen ist, ist im Fußgänger-Szenario, für niedrige Werte von q_{rel}, der Anteil der gesparten Ressourcen höher, als der Anteil des Qualitätsverlusts. Mit zunehmenden q_{rel} kehrt sich dieses Verhältnis jedoch um, so dass die Einsparung an Ressourcen geringer ist, als der Qualitätsverlust. Beim Auto-Szenario ist der Anteil der eingesparten Ressourcen sogar dauerhaft geringer als der prozentuale Qualitätsverlust. Zur Verdeutlichung zeigen die nachfolgenden Tabellen die genauen Werte der Schaubilder im Vergleich. Wir beginnen mit den Werten für das Fußgänger-Szenario:

q_{rel}	% Wegpunkte	% Energieverbrauch	Ersparnis pro Geräte [mJ]
0.01	$73,9\%$	$64,3\%$	3 286
0.05	$91,4\%$	$87,5\%$	959
0.1	$96,5\%$	$95,6\%$	147
0.3	$98,3\%$	99%	-187
0.5	$98,6\%$	97%	-211
1	100%	100%	0

Die nachfolgende Tabelle stellt die Werte für den Fall des Auto-Szenarios vor:

q_{rel}	% Wegpunkte	% Energieverbrauch	pro Geräte [mJ]
0.01	60%	$25,3\%$	17 893
0.05	$75,2\%$	$36,7\%$	15 169
0.1	$79,7\%$	$46,9\%$	12 710
0.3	$81,7\%$	$70,2\%$	7 131
0.5	$82,5\%$	71%	6 956
1	100%	100%	0

Die Tabellen zeigen zusätzlich die durchschnittliche Energie, welche pro Mobilgerät eingespart werden konnte in mJ.



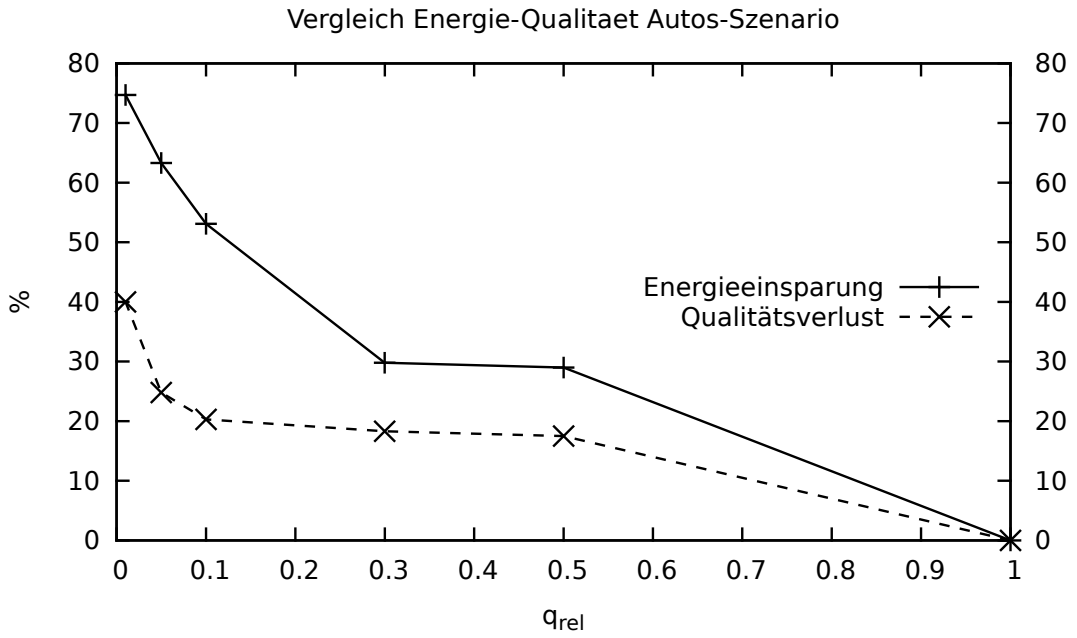


Abbildung 5.4: Vergleich Abnahme Qualität zu Einsparung Ressourcen

Damit wurden alle beabsichtigten Simulationen durchgeführt und die Ergebnisse können im folgenden Abschnitt zusammengefasst werden.

5.4 Zusammenfassung der Ergebnisse

Die Durchführung der Simulationen wurde in mehrere Schritten unterteilt, wobei wir damit begonnen haben die verschiedenen Szenarios hinsichtlich der Qualität zu untersuchen. Dabei haben wir festgestellt, dass die höhere Anzahl an GPS-Ortungen bei der trivialen Erfassung auch mit einer höheren Anzahl an Fusionen einhergeht, so dass viele der zusätzlichen GPS-Ortungen lediglich bereits erfasste Straßen beschreiben und nicht zur Entdeckung von neuen Straßen beitragen.

Im Anschluss wurde der Energieverbrauch der Mobilgeräte analysiert und dabei gezeigt, dass der zusätzliche Ressourcenverbrauch, der bei der gezielten Erfassung durch die Adhoc- bzw. Serverkommunikation entsteht, durch die niedrigere Anzahl an GPS-Ortungen, im Vergleich zur gezielten Erfassung, kompensiert wird, so dass die gezielte Erfassung insgesamt weniger Ressourcen verbraucht.

Bei der Gegenüberstellung der beiden Resultate wurde gezeigt, dass es möglich ist den relativen Energieverbrauch der beteiligten Mobilgeräte so zu reduzieren, dass dieser geringer ist als der relative Anteil des entstandenen Qualitätsverlusts. Unter Betrachtung der vorgestellten Ergebnisse, kann dieser Effekt so erklärt werden: Im trivialen Fall werden mit fortlaufender Zeit überwiegend Straßen erfasst, die bereits im Straßenmodell vorhanden sind. Bei der gezielten Erfassung werden Gitterzellen, die im Verhältnis zu den anderen Zellen qualitativ gut erfasst wurden, von der Erfassung ausgesetzt, so dass eine Einsparung an GPS-Ortungen entsteht.

Im Ergebnis ist die Einsparung höher als die zusätzliche Qualität die der triviale Fall generiert hat, da die Doppelerfassung von Straßen nur die Genauigkeit der bereits vorhandenen Straßen verbessert aber keine neuen Weginformationen für die Karte liefert. Dadurch konnte die Energie pro Geräte je nach Szenario um maximal 35% bei Fußgänger bzw. 75% bei Autos, im Vergleich zur trivialen Erfassung, gesenkt werden.

Betrachten wir die Unterscheide bei den Ergebnissen für das Fußgänger- bzw. Autoszenario, so können wir sehen, dass ein Szenario, in dem sich nur Autos befinden, eine höhere Ersparnis an Ressourcen bei der Erfassung ermöglicht, da die Anzahl der durchgeführten Fusionen viel höher ist als im Fußgänger-Szenario. Daher wollen wir kurz den Unterschied dieser beiden Szenarios analysieren: Da Autos auf Straßen fahren, sind ihre Bewegungsabläufe durch diese Straßen vordefiniert und daher untereinander relativ ähnlich. Sobald einen Straße von mehreren Autos befahren wurde, steigt ihr Qualitätswert, im Zuge der

durchgeführten Interpolationen, schnell an und der zugehörige Bereich wird von der Erfassung ausgesetzt. Anders sieht das Bewegungsverhalten von Fußgängern aus. Dieses ist vielfältiger, da es nur durch den Verlauf einer Straße begrenzt ist, aber nicht festgelegt ist, wie die Fußgänger die Straße passieren. Die Anzahl der durchgeführten Fusionen ist in diesem Fall geringer und daher werden weniger Teilbereiche von der Erfassung ausgesetzt als im Auto-Szenario. Daher zeigt sich, dass die gezielte Erfassung umso effektiver ist, desto homogener die Bewegungsabläufe der Mobilgeräte für eine feste Straße sind.

Kapitel 6

Zusammenfassung

Ziel dieser Masterarbeit war es Methoden und Konzepte zu finden, die eine automatische Kartenerstellung mit Hilfe von Mobilgeräten ressourcenschonend ermöglichen. Nach einer Heranführung an das Thema in den Grundlagen, sowie einer Analyse der bisherigen Arbeiten, welche Teilaspekte dieses Problems behandeln, wurde eine Reihe von Konzepten entwickelt, welche anschließend in einer Programmiersprache abgebildet wurden. Dadurch konnten diese Konzepte im Rahmen einer Simulation getestet und validiert werden.

Auch wenn es bereits eine Reihe von Forschungsarbeiten gibt, die sich damit beschäftigen, wie aus einer Menge von GPS-Tracks eine Straßenkarte generiert werden kann, so ist dies die erste Arbeit, welche sich mit dem konkreten Prozess der Erfassung dieser Daten selbst beschäftigt. Aus dieser Motivation heraus wurde ein Konzept zur gezielten Erfassung von GPS-Daten entwickelt, welches eine Ressourceneinsparung der Mobilgeräte ermöglicht. Diese gezielte Erfassung wurde in mehreren Teilschritten realisiert. Auf Basis der bereits vorhandenen Vorarbeiten, im Bereich der Generierung von Straßenkarten, wurde ein Verfahren definiert, welches die GPS-Daten der Mobilgeräte automatisch erfasst und in einem Straßenmodell integriert. Um die Qualität dieser bereits erfassten Straßen zu beurteilen, wurde anschließend eine Qualitätsmetrik für Straßen entwickelt. Mit deren Hilfe konnten ein Erfassungsalgorithmus entwickelt werden, der die Mobilgeräte anweist, bereits gut erfasste Teilbereiche von der Erfassung auszuschließen, um die Ressourcen der in diesem Teilbereich befindlichen Geräte zu sparen.

Durch Implementierung und Evaluation der in dieser Arbeit vorgestellten Ideen, konnte gezeigt werden, dass grundsätzlich eine Energieeinsparung erzielt werden konnte, die höher ist, als die dadurch entstandene Verminderung der Qualität der erstellten Straßenkarte. Somit wurde ein erster Schritt für weitere Betrachtungen dieser Problematik gemacht und die Grundlage für weitere Verbesserungen der vorgestellten Ergebnisses gelegt. In welche Richtungen diese gehen können, soll der folgende Abschnitt zeigen.

6.1 Ausblick

Das Ergebnis der vorliegenden Arbeit kann als erste Grundlagenarbeit, zur Definition eines Erfassungsalgorithmus für die automatische Generierung von Straßenkarten, gesehen werden. Wie bereits bei den Optimierungspotentialen in Abschnitt 3.4.10 angedeutet wurde, gibt es mehrere Ansätze, welche ein weiteres Einsparen von Ressourcen ermöglichen. So entstanden während der Erstellung dieser Arbeit eine Reihe von weiteren Ideen, die als Ausgangspunkt für künftige Weiterentwicklungen genutzt werden können. Nachfolgend sollen einige dieser kurz erläutert werden.

- Wie bereits in Abschnitt 3.3.2 erläutert wurde, könnte die Erfassung auch auf Ebene der einzelnen Straßen gesteuert werden, wenn ein Mobilgerät bei Betreten einer Straße, die im Straßenmodell auf dem Server bereits vorhanden ist, über deren Qualitätswert informiert werden könnte und daher die Erfassung für diese Straße komplett aussetzten würde, falls diese schon gut erfasst ist. Um dies realisieren zu können, müsste ein effizientes Vergleichsverfahren für Tracks gefunden werden, welches bereits anhand weniger Wegpunkte entscheiden kann, ob beide Tracks die gleiche Straße beschreiben. Um diese Entscheidung zu treffen, müsste das Mobilgerät aber Zugriff auf die Daten des Straßenmodells haben, so dass zusätzliche Kommunikationskosten anfallen würden. Ist so ein Verfahren gefunden, könnte man untersuchen, ob sich dieser Mehraufwand an Kosten über die Erfassungszeit wieder ausgleicht und insgesamt mehr Ressourcen eingespart werden könnten.

- Der in Abschnitt 3.3.2 eingeführte Zeitwert der Erfassung hat die Aufgabe den Qualitätswert der einzelnen Gitterzellen in Abhängigkeit zur verstrichenen Zeit zu dämpfen. Um diesen Zeitwert genauer an den tatsächlichen Stand der Erfassung anzupassen, kann untersucht werden, ob dieser gezielt durch einzelne Parameter gesteuert werden kann. Ein Beispiel wäre den Zeitwert mit einem Faktor zu multiplizieren, der die Größe des Zielbereichs im Verhältnis zur Anzahl der darin befindlichen Mobilgeräte berücksichtigt. Anschließend könnte diese Definition durch einen Vergleich mit dem tatsächlichen Fortschritt der Erfassung, im Rahmen einer Evaluation, überprüft werden.

- Um realistischere Ergebnisse zu erzielen, könnten die Simulationen, welche in der Evaluation getestet wurden, auf ein reales Szenario übertragen werden. Dazu werden Testpersonen mit Mobiltelefonen ausgestattet, auf denen die entsprechende Anwendung installiert ist. Diese führen diese Geräte im Alltag mit sich und zeichnen dabei GPS-Daten auf. Am Ende der Erfassung könnte der Energieverbrauch direkt von den Geräten abgelesen werden und die Genauigkeit der Erfassung mit vorhandenen Straßen verglichen werden.

- Ein wichtiger Faktor für das in dieser Arbeit vorgestellte Erfassungsverfahren, ist die in Abschnitt 3.3 definierte Qualitätsmetrik für Straßen. Dieses basiert auf den beiden Aspekten der Genauigkeit des Straßenverlaufs bzw. der Straßenposition, welche über die Winkel zwischen den Kanten der Wegpunkte bzw. über den Durchschnitt der Varianzen der Wegpunkte bestimmt ist. Dieses Modell könnte weiterentwickelt und auf allgemeine Akzeptanz geprüft werden, indem es zum Beispiel anhand empirischer Erhebungen evaluiert wird.

Neben diesen Überlegung gibt es noch weitere Ansatzpunkte, an denen zukünftige Arbeiten anschließen könnten. Dabei können die in der Evaluation erzielten Ergebnisse als Ausgang genommen werden, um einen Vergleich von Ergebnissen neuer Ansätze zu ermöglichen.

Literaturverzeichnis

[1] T. Abdelzaher, Y. Anokwa, P. Boda, J. Burke, D. Estrin, L. Guibas, A. Kansal, S. Madden, and J. Reich. Mobiscopes for human spaces. *Pervasive Computing*, 6(2):20–29, 2007.

[2] J. Agre and L. Clare. An integrated architecture for cooperative sensing networks. *Computer*, 33(5):106–108, 2000.

[3] E. Altmann and T. Jimenez. *NS Simulators for beginners, Lecture notes*. Univ. de Los Andes, Merida, Venezuaela and ESSI, Sophia-Antipolis, France, 2003-2004.

[4] AmsterdamRealtimeProject. http://realtime.waag.org, 26. Februar 2010.

[5] S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, K. Varadhan, Y. Xu, H. Yu, and D. Zappala. Improving simulation for network research. Technical Report 99-702b, University of Southern California, March 1999. revised September 1999, to appear in IEEE Computer.

[6] A. Bar-Noy, I. Kessler, and M. Sidi. Mobile users: to update or not to update? *Wirel. Netw.*, 1:175–185, June 1995.

[7] R. Bruentrup, S. Edelkamp, S. Jabbar, and B. Scholz. Incremental map generation with gps traces. In *International IEEE Conference on Intelligent Transportation Systems (ITSC), Vienna*. IEEE, 2005.

[8] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. Technical report, Center for Embedded Network Sensing, UC Los Angeles, 2006.

[9] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn. The rise of people-centric sensing. *Internet Computing, IEEE*, 12(4):12–21, July-Aug 2008.

[10] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson. People-centric urban sensing. In *WICON '06: Proceedings of the 2nd annual international workshop on Wireless internet*, page 18, New York, NY, USA, 2006. ACM.

[11] D. Chen, L. Guibas, J. Hershberger, and J. Sun. Road network reconstruction for organizing paths. In *ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[12] CitySense. http://www.citysense.com/, 5. Februar 2010.

[13] M. Conti. Multi-hop ad hoc networking: from theory to reality. In *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, pages 1–1, New York, NY, USA, 2007. ACM.

[14] D. Cuff, M. Hansen, and J. Kang. Urban sensing: out of the woods. *Commun. ACM*, 51(3):24–33, 2008.

[15] J. J. Davies, A. R. Beresford, and A. Hopper. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing*, 5:47–54, October 2006.

[16] C. de Boor. *A Practical Guide to Splines*. Springer-Verlag, 2001.

[17] T. Farrell, K. Rothermel, and R. Cheng. Processing continuous range queries with spatio-temporal tolerance. *Transactions on Mobile Computing*, 2010.

[18] P. Forbrig. *Objektorientierte Softwareentwicklung in UML*. Hanser Fachbuchverlag, 3 edition, 2007.

[19] E. M. Gillavry. Collaborative mapping: grassroots lbs. Presentation delivered during the first session "Location, location, location: cartography in a 3G environment" of Cartography 2003, 2.-5. September 2003.

[20] S. Gorman. Creating maps for everyone and network effects for the data driving them. *Receiver magazine*, 21, 2008.

[21] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global Positioning Systems, Intertial Navigations, and Integration*. John Wiley and Sons, Inc., 2001.

[22] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[23] N. Hönle, M. Grossmann, D. Nicklas, and B. Mitschang. Preprocessing position data of mobile objects. In *MDM '08: Proceedings of the The Ninth International Conference on Mobile Data Management*, pages 41–48, Washington, DC, USA, 2008. IEEE Computer Society.

[24] S. Y. Hui and K. H. Yeung. Challenges in the migration to 4g mobile systems. *IEEE Communications Magazine*, 41(12):54 – 59, December 2003.

[25] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. *The International Series in Engineering and Computer Science*, 535:153–181, 1996.

[26] E. Kanjo, J. Bacon, D. Roberts, and P. Landshoff. Mobsens: Making smart phones smarter. *IEEE Pervasive Computing*, 8(4):50–57, 2009.

[27] C. Kechers. *UML 2: Das umfassende Handbuch.* Galileo Computing, 3 edition, 2009.

[28] J. Kim, V. Sridhara, and S. Bohacek. Realistic mobility simulation of urban mesh networks. Technical report, Department of Electrical and Computer Engineering, University of Delaware, 2007.

[29] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[30] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell. Urban sensing systems: opportunistic or participatory? In *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 11–16, New York, NY, USA, 2008. ACM.

[31] R. Lange, T. Farrell, F. Durr, and K. Rothermel. Remote real-time trajectory simplification. In *PERCOM '09: Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–10, Washington, DC, USA, 2009. IEEE Computer Society.

[32] R. Lange, H. Weinschrott, L. Geiger, A. Blessing, F. Dürr, K. Rothermel, and H. Schütze. On a generic uncertainty model for position information. In *Proceedings of the 1st International Workshop on Quality of Context (QuaCon '09)*, pages 76–87, Stuttgart, Germany, June 2009.

[33] A. Leonhardi and K. Rothermel. A comparison of protocols for updating location information. Technical report, Institute of Parallel and Distributed High-Performance Systems, Department of Computer Science, University of Stuttgart, March 2000.

[34] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley. Mobility improves coverage of sensor networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 300–308, New York, NY, USA, 2005. ACM.

[35] N. Meratnia and R. de By. Spatiotemporal compression techniques for moving point objects. *Proceedings of the 9th. International Conference on Extending Database Technology*, pages 765 – 782, March 2004.

[36] MIT. The cartel project. http://cartel.csail.mit.edu, 18. Februar 2010.

[37] S. Morris, A. Morris, and K. Barnard. Digital trail libraries. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 63–71, New York, NY, USA, 2004. ACM.

[38] OpenStreetMap. http://www.openstreetmap.org, 24. Februar 2010.

[39] O. Riva and C. Borcea. The urbanet revolution: Sensor power to the people! *IEEE Pervasive Computing*, 6(2):41–49, 2007.

[40] C. Rizos. Principles and practice of gps surveying. Technical report, School of Geomatic Engineering - The University of New South Wales, 1999.

[41] J. Roth. Extracting line string features from gps logs. *Schriftenreihe der Georg-Simon-Ohm-Hochschule Nuernberg*, 2008.

[42] A. Sant. Redefining the basemap. *intelligent agent*, 6(2):99 – 105.

[43] C. M. U. Sensing. http://www.escience.cam.ac.uk/mobiledata/, 23. April 2010.

[44] R. Silverstone and Z. Sujon. Urban tapestries: Experimental ethnography, technological identities and place. Electronic Working Papers No. 7, Media@lse, Department of Media and Communications, University of London, 2005.

[45] M. Story and R. G. Congalton. Accuracy assessment: A user's perspective. *Photogrammetric Engineering and Remote Sensing*, 52(3):397 .. 399, March 1986.

[46] Tagzania. http://www.tagzania.com/, 24. Februar 2010.

[47] H. Weinschrott, F. Durr, and K. Rothermel. Efficient capturing of environmental data with mobile rfid readers. In *MDM '09: Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 41–51, Washington, DC, USA, 2009. IEEE Computer Society.

[48] J. Yoo and C. Kim. On the hidden terminal problem in multi-rate ad hoc wireless networks. In *Information Networking*, volume 3391, pages 479 – 488. Springer Berlin / Heidelberg, 2005.